FINITE ELEMENT ANALYSIS PROGRAM (FEAP) FOR CONDUCTION HEAT TRANSFER

Jorge Martins Bettencourt

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

FINITE ELEMENT ANALYSIS PROGRAM (FEAP)

FOR CONDUCTION HEAT TRANSFER

by

Jorge Martins Bettencourt

December 1979

Thesis Advisor: G. Cantin

Approved for public release; distribution unlimited

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Finite Element Analysis Program (FEAP) for Conduction Heat Transfer		5. TYPE OF REPORT & PERIOD COVERED Master's & Engineer's Thesis December 1979
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jorge Martins BETTENCOURT		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE December 1979
		13. NUMBER OF PAGES 161
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)* Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. *(of this report)* Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Conduction Heat Transfer, Finite Element, Numerical Analysis, Time Integration Operator, Computer Programming

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

    The Finite Element Analysis Program (FEAP) was expanded to solve linear and nonlinear, two and three dimensional heat conduction problems. The usual types of boundary conditions, including radiation, may be specified. A wide range of two- and three-time level schemes for the solution of time dependent problems is available in the program and a discussion of those most commonly used is presented. The algorithms for the solution of typical

DD FORM 1473 1 JAN 73   EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |

practical problems are described and several numerical examples are presented. The results are compared with the available analytical solutions. A listing of this expanded version of FEAP and the corresponding user's instructions are provided.

Finite Element Analysis Program (FEAP)
for Conduction Heat Transfer

by

Jorge Martins Bettencourt
Lieutenant, Portuguese Navy
B.S., Naval Postgraduate School, 1979

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

and

MECHANICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
December 1979

ABSTRACT

The Finite Element Analysis Program (FEAP) was expanded to solve linear and nonlinear, two and three dimensional heat conduction problems. The usual types of boundary conditions, including radiation, may be specified. A wide range of two- and three-time level schemes for the solution of time dependent problems is available in the program and a discussion of those most commonly used is presented. The algorithms for the solution of typical practical problems are described and several numerical examples are presented. The results are compared with the available analytical solutions. A listing of this expanded version of FEAP and the corresponding user's instructions are provided.

TABLE OF CONTENTS

5

ACKNOWLEDGMENTS

The author wishes to express his sincere appreciation to Professor Gilles Cantin for is invaluable guidance, help and friendship as instructor and thesis advisor. In addition, the author wishes to thank Professor Olgierd Cecil Zienkiewicz for his interest, comments and thoughtful advice.

The author is obligated to Professor Matthew Kelleher and Professor David Salinas for their assistance during the first stages of this work.

Finally the author wishes to thank his wife, Maria João, and his daughters, Joana and Catarina, for their understanding and encouragement throughout the course of this work.

I. INTRODUCTION

This thesis describes an application of the Finite Element method to Heat Transfer analysis through the use of a computer program. This program was designed to solve linear and nonlinear, steady and unsteady, two and three dimensional heat conduction problems involving temperature dependent thermophysical properties and complicated radiation/convection boundary conditions.

The Finite Element Analysis Program (FEAP), programmed by Professor R. L. Taylor in the Department of Civil Engineering of the University of California, Berkeley, served as the point of departure for the present code. It has now been expanded with two additional modules for Heat Transfer analysis and time integration of first order equations, respectively, but the original characteristics of a research and educational tool in which the various modules can be changed or added to as desired, were maintained.

Although all equations are retained in core, the program can handle realistic engineering problems with several hundred unknowns in most computer systems.

While the algorithms to solve steady heat transfer problems are well studied and defined, more research has to be done on the solution of unsteady problems. The Zienkiewicz two- and three-time level schemes were integrated in the program in such a way that a wide range of

choices of time integration algorithms is available to the user. A limited study was performed in order to determine the characteristics of the most used members of those families of numerical schemes.

The FEAP code is written in FORTRAN IV language and this version was constructed and tested on an IBM 360/67 computer system with OS/67 release 18. Other systems and installations will require modifications.

II. STATEMENT OF OBJECTIVES

A. EQUATION OF CONDUCTION. INITIAL AND BOUNDARY
CONDITIONS

The problem considered for solution is thoroughly developed by Arpaci in Ref. 1 and is mathematicaly described in the region Ω by the equation

$$\rho\, c\, \frac{\partial T}{\partial t} = \overline{\nabla} \cdot (\overline{k}\, \overline{\nabla}T) + Q \qquad (1)$$

subjected to boundary conditions

$$T = T_w \qquad \text{on} \qquad \Gamma_1 \qquad (2)$$

and

$$(\overline{k}\, \overline{\nabla}T)\, \overline{n} + q + q_c + q_r = 0 \qquad \text{on} \qquad \Gamma_2 \qquad (3)$$

and initial condition

$$\lim_{t \to 0} T = T_0 \qquad (4)$$

$\overline{\nabla}$ is the gradient operator, Γ_1 and Γ_2 are mutually exclusive parts of the boundary of the region Ω, T is the temperature and t is the time. The thermal capacity ρc, the thermal conductivity $\overline{k}$ and the rate of internal heat generation Q are thermophysical properties dependent on temperature.

In equation (3) $\overline{n}$ is an outward unit vector normal to the boundary surface while q, q_c, q_r represent respectively the imposed heat flux and the rates of heat flow per unit area due to convection and radiation defined as

$$q_c = h_c (T - T_{ac}) \qquad (5)$$

and

9

$$q_r = F\sigma (T^4 - T_{ar}^4) = h_r (T - T_{ar}) \qquad (6)$$

In (5) h_c is the temperature dependent convective heat transfer coefficient. In (6) the parameter h_r is defined by the expression

$$h_r = F\sigma (T^2 + T_{ar}^2) (T + T_{ar}) , \qquad \text{where } F \text{ is}$$

the radiant exchange factor and σ the Stefan-Boltzman constant. T_{ac} and T_{ar} are the equilibrium temperatures for which, respectively, no convection and radiation occurs.

B. NUMERICAL SOLUTION. DISCRETIZATION BY THE GALERKIN METHOD. MATRIX FORMULATION

The spacewise discretization of equation (1) using cartesian coordinates can be acomplished by Galerkin's principle as shown by Zienkiewicz in Ref. 2 and Lew in Ref. 3.

Let the unknown function T be approximated, throughout the solution domain at any time t, by the relationship

$$T = \sum_{i=1}^{n} N_i(x,y,z) \, T_i(t) = <N> \{T\} \qquad (7)$$

where N_i are the usual shape functions defined piecewise element by element, T_i being the nodal parameters.

The result is

$$[K] \{T\} + [C] \{\dot{T}\} + \{F\} = \{0\} \qquad (8)$$

where [K] and [C] are symmetric matrices defined, on the element level, as

$$[K]^e = \int_{\Omega^e} \{ <\frac{\delta N^T}{\delta x}> <\frac{\delta N}{\delta x}> k_x + <\frac{\delta N^T}{\delta y}> <\frac{\delta N}{\delta y}> k_y +$$
$$<\frac{\delta N^T}{\delta z}> <\frac{\delta N}{\delta z}> k_z \} \, d\Omega +$$
$$\int_{\Gamma_2^e} (h_c + h_r) <N>^T <N> \, d\Gamma \qquad (9)$$

10

$$[C]^e = \int_{\Omega^e} \rho c \langle N \rangle^T \langle N \rangle \ d\Omega \qquad (10)$$

The load vector F is

$$\{F\}^e = -\int_{\Omega^e} \langle N \rangle^T Q \ d\Omega +$$
$$\int_{\Gamma_2^e} \langle N \rangle^T (q - h_c T_{ac} - h_r T_{ar}) \ d\Gamma \qquad (11)$$

where the surface integrals are performed only over the
surfaces where the prescribed boundary condition applies.

It must be noted that the set of equations (8) is
nonlinear since the matrices [K] , [C] and vector {F}are
dependent on T .

The system of equations (8) may now be solved by any
automated numerical technique. Code FEAP was written for
this purpose and is described in the rest of this thesis.

11

iii.   <u>FINITE ELEMENT ANALYSIS PROGRAM</u>

The Finite Element Analysis Program (FEAP) was programmed and published by Taylor in Ref. 4 where it is well explained. The reader should consult this reference for details about it.

This thesis is an expansion of the original program towards the solution of Heat Conduction problems and as such only the main features of the basic program will be referred in this section. A more detailed explanation is reserved for the new subroutines added to the program. The complete user's instructions for FEAP are given in Appendix A and they complement this section.

.

A.   MAIN PROGRAM

FEAP can be separated into two basic parts:

a) Data input module and preprocessor

b) Solution and output modules

The data input module must transmit sufficient information to the other modules so that each problem can be solved. All the input data is stored in a single array (integer array M) which is partitioned to store all the data arrays, as well as the global arrays, e.g., stiffness, mass, load, etc.

The total capacity of the program is controlled by the dimension of the array M in the blank common of the main

program and the corresponding value of the variable MAX.

The partition of M is performed in the control routine (subroutine PCONTR) and the data used for it is supplied by the user in the title and control information cards (first two cards for every run of the program).

Then the nodal coordinates, element connections, material properties, nodal loading and boundary conditions may be input using the macro control statements. They are interpreted by subroutine PMESH and each control is a function independent of the others.

To clarify what was said till this point, the discussion of a simple example may be useful. Consider the mesh presented in [1]Figure 1 and assume that quadratic rectangular elements as the one defined in Figure 2 are used. This example is completly solved in Appendix B and will be used throughout this text.

The sector of a circular ring was divided in six elements and the 33 nodes were numbered. The order of numbering is not crucial but in order to improve the profile of non-zero coefficients the following general rule should be used.

The numbering should be such as to minimize the nodal difference for each element (maximum node nummber minus minimum node number).

The user can now proceed to the preparation of data for the program. The first step consists of specifying the problem title and the control information. The latter is

[1]Figures are grouped at the end pp 47-79

clearly 33 nodes, six elements, one material set, two spatial dimensions and eight nodes per element. If this is a Heat Transfer problem we have one unknown per node (temperature).

The program expects now the data cards for mesh description. Any analysis require at least

a) coordinate data which follows the macro command COOR

b) connectivity table which follows the macro command ELEM

c) material data which follows the macro command MATE

In addition, most analysis will require specification of nodal boundary restraint conditions, macro BOUN, and the corresponding nodal force or displacement values, macro FORC. The term displacement refers to the value of the unknown for each specific problem. In a Heat Transfer problem the unknown is temperature while in an Elasticity analysis it is a geometrical displacement.

The TEMP macro command (temperature data) shall not be used in a Heat Transfer analysis to specify nodal temperature. It may be used to specify auxilary nodal quantities.

If in our problem we assume the line defined by nodes 31, 32 and 33 at the temperature of 20, the line defined by nodes 1, 2 and 3 at 820 and insulated elsewhere, the macro control statements and correspondent data cards will be

COOR

1   5   .1

14

```
  31          .3
   4      5.116666667
  29        .283333333
   2    5  .1           3.
  32          .3         3.
   3    5  .1           6.
  33          .3         6.
   5      5.116666667  6.
  30        .283333333  6.
   (blank card)
POLA
   1   33    1
   (blank card)
ELEM
   1    1    1    6    8    3    4    7    5    2    5
   (blank card)
BOUN
   1    1   -1
   3        -1
  31    1   -1
  33        -1
   (blank card)
FORC
   1    1  820.
   3       820.
  31    1   20.
  33       20.
```

```
     (blank card)
MATE

     1     2
10.        10.         8000.      250.                               3
END
```

In spite of the fact that the MATE command must be
included in this part of the data preparation, its
discussion is left to the section where the element modules
are treated.

After completing the mesh data input, we are ready to
initiate the problem solution.

FEAP has modules for variable algorithm capabilities and
which, if necessary, can be modified or expanded. The basic
aspect of the variable algorithm program is a macro
instruction language which can be used to construct modules
for specific algorithms as needed. This language is
interpreted by subroutine PMACR and the complete list of
macro instructions available in this version of FEAP is
given in Appendix A.3.

Here, as an example, we will discuss in detail the
algorithm for the solution of linear steady state problems
using the original explanation given in Ref. 4.

To use the macro programming commands, the user only
needs to learn the mnemonics of the language. If one wishes
to form the global stiffness matrix the program instruction
TANG is used (TANG is the mnemonic for a symmetric tangent
stiffness matrix and for nonlinear elements would form and

assemble into the global stiffness the element tangent
stiffness computed about the current displacement state; for
linear elements this is just the linear stiffness matrix).
For a problem with an unsymmetric tangent stiffness the
macro command UTAN is used.

If one wishes to form the right hand side of the
equations modified for specified displacements one uses the
program instruction FORM. The resulting equations are
solved using the instruction SOLV.

Printed output can be obtained using the macro command
DISP.

The above instructions are sufficient to solve linear
steady state problems, that is, the macro instructions
TANG (or UTAN)
FORM
SOLV
DISP
are precisely the required instructions to solve any linear
steady state problem.

If the same block of instructions is repeated twice or
more, considerable effort is wasted in preparing the macro
instruction data. To rectify this, looping commands are
introduced as the instruction pair
LOOP   n
   .
   .
   .
NEXT
which indicates that looping over all instructions between

17

LOOP and NEXT will occur n times.

Many other classes of problems can be solved using the macro instruction list given in Appendix A. With the above short explanation and the discussion presented in section IV for Heat Transfer problems, the reader will be able to construct his own algorithms using the macro programming language.

B. FINITE ELEMENT SOLUTION MODULES

Most of FEAP is common to a wide range of problems, but some of the computations are linked to the type of problem to be solved. It is clear that if one wishes to solve an Elasticity problem, the calculation of the stiffness matrix will be different from that used for a Heat Transfer problem. It is also clear that differences also arise if one is using triangular elements instead of quadratic isoparametric elements as a way of describing the continuum.

The program was designed such that all computations associated with any type of element are contained in an element subroutine called ELMTnn where nn is between 01 and 05 in this version of FEAP. Each element type to be used is specified as part of the material property data, following macro control MATE.

Since the objective of this work is the solution of Heat Transfer problems, the element modules discussed here are the two and three dimensional heat tranfer modules named ELMT02 and ELMT03.

1.   Two Dimensional Heat Transfer Element

Subroutine ELMT02 is accessed using the number two in column ten of each material number card.

According to the value of the variable ISW, defined in subroutine PMACR, a specific function is performed. If ISW=1 it reads and prints the material property data and line boundary conditions. The stiffnes and mass matrices are computed if ISW=3 and ISW=5 respectively. The load vector due to internal forces and boundary conditions is calculated when ISW=6.

The material properties may be input as constants or as a table of temperature and property values. If any other way of defining the property values is required by a specific problem, subroutines PKX , PKY , PROC or PQ which calculate respectively the conductivity in x and y directions, the thermal capacity and the heat generation per unit volume, can be easily modified or substituted.

The shape functions used by this heat tranfer module are calculated by subroutines SHAPE and SHAP2. These routines are capable of constructing shape functions for a three-node triangle, a four-node linear quadrilateral, an eight-node quadratic serendipity quadrilateral, a nine-node quadratic Lagrangian quadrilateral or any combination in-between.

A four- to nine-node two dimensional element is shown in Figure 3. As shown by Bathe and Wilson in Ref. 5 the temperature within the element is expressed at any time

19

in the local coordinate system s, t in terms of the nodal temperatures T_i by

$$T(s,t) = \Sigma\ N_i(s,t)\ T_i$$

where

$$N_1 = (1-s)(1-t)/4 - (N_5+N_8)/2 - N_9/4$$
$$N_2 = (1+s)(1-t)/4 - (N_5+N_6)/2 - N_9/4$$
$$N_3 = (1+s)(1+t)/4 - (N_7+N_6)/2 - N_9/4$$
$$N_4 = (1-s)(1+t)/4 - (N_7+N_8)/2 - N_9/4$$
$$N_5 = (1-s^2)(1-t)/2$$
$$N_6 = (1+s)(1-t^2)/2$$
$$N_7 = (1-s^2)(1+t)/2$$
$$N_8 = (1-s)(1-t^2)/2$$
$$N_9 = (1-s^2)(1-t^2)$$

If any of the nodes from five to nine are omitted the corresponding value of N is zero.

The connectivity table must follow the numbering convention shown in Figure 3, otherwise wrong results will be obtained.

The maximum number of nodes per element will depend on the type of elements to be used and may be from three to nine. The eight-node serendipity shape functions occur if the ninth node number is omitted. The four-node quadrilateral shape functions are computed if only the first four nodal connections are non zero and the three-node triangle if the first three nodal connections are non zero. Finally, if a mid-side nodal connection is omitted the edge is linear.

The line boundary conditions, closely related to the type of element used, are defined in the cards following the macro MATE. The line subjected to a specified boundary condition is identified using the local coordinates. A line is numbered 1 or 2 if it is perpendicular to the axis s or t, respectively. Those numbers are positive or negative according to which direction of the axis it is perpendicular. Thus, as an example, the line defined by the nodes 2, 6 an 3 is numbered 1 while the line defined by nodes 1, 5 and 2 is numbered -2.

The boundary conditions are treated in subroutine BCOND2 and four types are considered:

a) specified flux

b) convection with constant heat tranfer coefficient

c) convection with temperature dependent heat transfer coefficient

d) radiation

If more complex boundary conditions are required subroutine BCOND2 may be easily modified.

The numerical integration necessary to evaluate the matrices and vectors in equation (6) are performed using the Gauss quadrature formula; the abscissas and weight coefficients are calculated in subroutine PGAUSS. The user may choose the number of points per direction used in the integration from one to six, but the default value is four points .

To illustrate the above, we assume that our sector

21

used as an example is part of the cross section of a hollow
cylinder made of a material with conductivity in x and y
directions of 10, specific heat 250 and density 8000. If the
outside surface is subjected to convection to an ambient
atmosphere at 20 degrees with a constant heat tranfer
coefficient of 200, the cards following the macro card MATE
are   .

MATE

```
    1    2
10.      10.        8000.     250.                      3
    6    2    1  200.       20.
```

It must be noted that a plane analysis is specified
and the integration will be performed using three points per
direction. The reader must also note that this version of
the problem is slightly different from the one given when
the mesh data preparation was discussed. In the previous
problem the line now subjected to a convection boundary
condition was at a specified constant temperature of 20. If
this second problem is to be solved the cards corresponding
to nodes 31-33 in the BOUN and FORC sets must be omitted.

Following is a list of subroutines included in the
two dimensional heat tranfer module:

ELMT02 : main routine; forms the necessary arrays
for the solution.

SHAPE : calculates the shape functions for the
triangle and the linear quadrilateral elements.

SHAP2 : adds the quadratic terms and the center node

22

to the shape functions.

PGAUSS : determine the abscissae and weight coefficients for the numerical integration.

BCOND2 : adds to the arrays calculated in ELMT02 the contribuition from the specified boundary conditions.

PKX : determine the temperature dependent conductivity in the x direction.

PKY : determine the temperature dependent conductivity in the y direction.

PROC : determine the temperature dependent thermal capacity.

PQ : determine the temperature dependent heat generation per unit volume.

CONV : determine the temperature dependent heat transfer coefficient.

TABLE : called by PKX, PKY, PQ, PROC and CONV; calculates the correspondent coefficient to a given temperature by linear interpolation between two consecutive entries in a table.

JACBB2 : calculates the jacobian determinant when an integration over a line is necessary.

2. Three Dimensional Heat Transfer Element

This module is called ELMT03 and is accessed using the number three in column ten of each material number card.

This element is the generalization of ELMT02 for three dimensional space and little remains to be said about its use.

The material properties are read in a similar way as the two dimensional case, but the conductivity in the z direction must be added. This module uses the same subroutines as ELMT02 to read and calculate the temperature dependent properties.

Subroutines SHAP3D and SHAP3 may construct interpolation functions for any combination between a eight-node linear brick and a 21-node Lagrangian brick.

An eight- to twenty one-node three dimensional solid element is shown in Figure 4. The natural coordinates (r, s and t) of the eight corner nodes are $(\pm 1, \pm 1, \pm 1)$, of the twelve mid-edge nodes are $(0, \pm 1, \pm 1)$, $(\pm 1, 0, \pm 1)$ and $(\pm 1, \pm 1, 0)$ and of the center node are $(0, 0, 0)$.

The temperature within the element is defined in terms of the nodal temperatures T_i at any time by [Ref. 5]

$$T(r,s,t) = \Sigma \ N_i(r,s,t)T_i$$

If we define

$$G(\beta, \beta_i) = .5(1+\beta_i\beta) \ , \quad \text{for } \beta_i = \pm 1$$
$$= 1-\beta^2 \quad \text{for } \beta_i = 0$$

and

$$g_i = G(r,r_i)G(s,s_i)G(t,t_i)$$

where r_i, s_i and t_i are the natural coordinates of the element nodal points, the interpolation functions N_i are

$$N_1 = g_1 - (N_9 + N_{12} + N_{17})/2 - N_{21}/8$$
$$N_2 = g_2 - (N_9 + N_{10} + N_{18})/2 - N_{21}/8$$
$$N_3 = g_3 - (N_{10} + N_{11} + N_{19})/2 - N_{21}/8$$
$$N_4 = g_4 - (N_{11} + N_{12} + N_{20})/2 - N_{21}/8$$

$$N_5 = g_5 - (N_{13}+N_{16}+N_{17})/2 - N_{21}/8$$

$$N_6 = g_6 - (N_{13}+N_{14}+N_{18})/2 - N_{21}/8$$

$$N_7 = g_7 - (N_{14}+N_{15}+N_{19})/2 - N_{21}/8$$

$$N_8 = g_8 - (N_{15}+N_{16}+N_{20})/2 - N_{21}/8$$

$$N_j = g_j - N_{21}/4 \qquad \text{for } j=9,\ldots,20$$

$$N_{21} = g_{21}$$

If any of the nodes from nine to twenty one are omitted the corresponding value of g is zero.

The numbering convention used in the definition of the shape functions and shown in Figure 4 must be followed by the user when the connectivity table is built and the surface boundary conditions are specified. The surfaces are numbered ± 1, ± 2, ± 3 as they are perpendicular to the positive or negative directions of the axis r, s, t, respectively.

As the numerical integration routine is the same as for ELMT02, the user may choose from one to six points per direction, the default value being four points.

Following is a list of subroutines included in the three dimensional heat transfer module:

ELMT03 : main routine; forms the matrices necessary for the solution.

SHAP3D : calculates the shape functions for the eight node linear brick.

SHAP3 : adds the quadratic terms to the shape functions and the center node for the Lagrangian brick.

BCOND3 : adds to the matrices the contribuition from

the specified boundary conditions.

PKZ : determine the conductivity in the z direction

JACBB3 : calculates the jacobian determinant when an integration over a surface is necessary.

C. TIME INTEGRATION MODULE

The first order ordinary differential equation solver is accessed when the macro command ODE1 is utilized and uses the Zienkiewicz two- and three-time level schemes. The details of these algorithms are treated in section IV.

This module was designed for the solution of first order equations but can, with little programming effort, be expanded in order to include higher order algorithms and solve higher order differential equations.

In order to economize computer memory space at some cost of execution time, this module only uses the space previously reserved for the mesh construction. As the mesh data is not needed during the execution of a time step integration, all the corresponding arrays are kept in file 9 during that period. Once the next displacement vector is calculated and the current displacement vector is saved in file 10, all the information in file 9 is retrieved and the mesh data restored.

File 10 is used as a working space during the time integration and the current displacement vector is saved in it for possible future use. This vector is retrieved if the algorithm used in the next time step integration is a three point scheme.

Besides the time step size change using the appropriate macro instructions (macro TOL), an optional automatic time step adjustment was incorporated in the program. The norm of the difference between the displacements vectors at two consecutive times is computed at each step. If the norm is less than a predetermined value ΔT_{max} , the time step size is doubled before going to the next step, whereas if the norm is greater than a prespecified value ΔT_{min}, the time step size is halved and the calculation for that time step is repeated until the norm is acceptable. The magnitudes of the maximum and minimum values of the norm are problem dependent and must be choosen by the user. If they are not specified no time step adjustment will be performed. The recalculation of the displacement vector is allways done using the two point scheme, even when a three point scheme was utilized for the first calculation.

The macro command ODE1 used to access this module must be followed by a second macro command in order to determine the function to be performed. The secondary macro instructions are INIT, LINE or QUAD.

The couple ODE1 INIT reads the integration constants theta , beta and gamma, the parameters for the automatic time step adjustment and the initial displacement vector. This data must follow the macro program (see user's instructions). No time integration is performed by this instruction.

The couple ODE1 LINE performs the two point scheme

27

and the current displacement vector is substituted by the new calculated displacement vector.

The couple ODE1 QUAD has a function similar to ODE1 LINE but uses the three point scheme.

Once the execution of an ODE1 macro command is complete, the mesh data is restored and the displacement at time t replaced by the displacement at time t+ Δt, when the secondary macro command is LINE or QUAD. If INIT is used, the displacement vector becomes the given initial displacement vector.

The subroutines included in this module are:

PODE1 : control routine; reserves working space and calls the appropriate subroutines.

INIT : reads the constants theta, beta and gamma and the maximum and minimum values for the norm used in the time step adjustment.

PLINE : performs the two point integration scheme.

PQUAD : performs the three point integration scheme.

TERM : performs the automatic time step adjustment.

IV. APPLICATION OF FEAP TO HEAT TRANFER PROBLEMS

The program FEAP requires from the user the knowledge of the algorithm to be used in the solution of the problem to be treated.

In this section the algorithms used in the solution of heat conduction problems are discussed.

A. STEADY STATE PROBLEMS

1. Algorithms for Linear Problems with Macro Program

In this case equation (8) becomes

$$[K] \{T\} + \{F\} = \{0\} \tag{12}$$

The algorithm used to solve (12) is described by

$$[K] \{v\} = -\{F\} - [K] \{T^\circ\} \tag{13}$$

$$\{T\} = \{T^\circ\} + \{v\} \tag{14}$$

where $\{T^\circ\}$ are the specified nodal temperatures.

The macro instruction TANG builds the left hand side of (13) while FORM forms the vector in the right hand side. The instruction SOLV solves the system (13) and performs the step defined by (14).

Consequently the simplest macro program used to solve this problem is

TANG

FORM

SOLV

DISP

29

2. Algorithms for Nonlinear Problems with Macro
    Program

Equation (12) still applies to this case but now $[K]$
and $\{F\}$ may be dependent on the nodal temperatures $\{T\}$. The
well known Newton-Raphson iteration may be used and the
algorithm is described as

$$[K(T^i)] \{v^i\} = -\{F(T^i)\} - [K(T^i)] \{T^i\} = \{R^i\} \quad (15)$$

$$\{T^{i+1}\} = \{T^i\} + \{v^i\} \quad (16)$$

where $\{T^i\}$ are the nodal temperatures at the ith iteration.
The algorithm defined by (15) and (16) is the one used for
the linear problem repeated several times. Then the macro
program may be

LOOP   n

TANG

FORM

SOLV

DISP

NEXT

DISP

where n are the user's guess of the number of iterations
necessary to obtain equilibrium. However, the program has an
internal check on the value of the vector norm $||R^i||$,
where

$$||R|| = (\sum_{k=1}^{n} |R_k^i|^2)^{1/2}$$

Whenever

$$||R^i|| < \text{TOL} \times \max_j ||R^j|| \quad (j=1,\ldots,i)$$

where TOL is a predefined tolerance (10^{-9} is the default

value), the iteration ceases and a skip to the macro command immediately following the first NEXT occurs. Usually one begins with zero as the initial guess of the displacement vector; however, any other vector may be used.

B.   TIME DEPENDENT PROBLEMS

1.   Two and Three Point Recurrence Schemes for First Order Equation

The set of differential equations

$$[K]\{T\} + [C]\{\dot{T}\} + \{F\} = \{0\} \tag{8}$$

may be solved using one of the many recursive schemes described in the literature. From them, the two and three time level schemes presented by Zienkiewicz [Ref.4] offer a wide range of choices for the solution of linear and nonlinear problems.

The recurrence relation for the two-time level scheme can be written as

$$(\frac{1}{\Delta t}[C] + \Theta[K])\{T_{n+1}\} + (-\frac{1}{\Delta t}[C] + (1-\Theta)[K])\{T_n\} +$$

$$\{\overline{F}\} = \{0\} \quad , \quad 0 \le \Theta \le 1 \tag{17}$$

where the subscripts n and n+1 denote evaluation at time t and t+Δt. The vector $\{\overline{F}\}$ is defined as

$$\{\overline{F}\} = \{F_{n+1}\} + (1-\Theta)\{F_n\} \tag{18}$$

The choice of the parameter Θ defines the particular scheme to be used and the reader will recognize a well known series of finite difference formulas with a modification of using a weighted loading term $\{\overline{F}\}$.

Consider the system of decoupled equations in terms of the modal participation variables T_i

31

$$c_i \, \dot{T}_i + k_i \, T_i + f_i = 0 \qquad (19)$$

for free response, i.e., $f_i=0$, expression (17) becomes

$$(\tfrac{1}{\Delta t} c_i + k_i \theta) \, (T_i)_{n+1} + (-\tfrac{1}{\Delta t} c_i + k_i(1-\theta)) \, (T_i)_n = 0 \qquad (20)$$

Substituting the relation

$$(T_i)_{n+1} = \lambda (T_i)_n \qquad (21)$$

in (20), the resulting characteristic equation of the recurrent scheme solved for λ gives

$$\lambda = [1 - k_i(1-\theta)\Delta t/c_i] / [1 + k_i \theta \, \Delta t/c_i]$$

The scheme will be unconditionally stable if

$$|\lambda| < 1$$

for any value of p_i where

$$p_i = (k_i/c_i) \, \Delta t \qquad (22)$$

This condition is satisfied for

$$\theta \geq 1/2$$

On the other hand if

$$0 < \theta < 1/2$$

stability is conditional requiring

$$p_i < 2/(1-2\theta)$$

Figure 5 shows how λ varies with p_i for the schemes discussed later and how it compares with the exact value of

$$\lambda = \exp(-p_i)$$

The schemes considered are

a) Crank-Nicolson with $\theta=1/2$

b) $\theta=2/3$ proposed by Zienkiewicz [Ref. 4]

c) $\theta=3/4$ proposed by the author

d) $\theta=0.878$ proposed by Liniger in Ref. 6.

The Zienkiewicz three-time level scheme is defined

32

as

$$(\gamma[C] + \beta\,\Delta t[K])\{T_{n+1}\} + ((1-2\gamma)[C] + (1/2-2\beta+\gamma)\Delta t[K])\{T_n\} +$$
$$(-(1-\gamma)[C] + (1/2+\beta-\gamma)\Delta t[K])\{T_{n-1}\} + \Delta t\{F\} = \{0\} \qquad (23)$$

where the subscripts n, n+1 and n-1 denote evaluation at time t, t+Δt and t-Δt respectively.

The vector $\{\overline{F}\}$ is defined as

$$\{\overline{F}\} = \beta\{F_{n+1}\} + (1/2-2\beta+\gamma)\{F_n\} + (1/2+\beta-\gamma)\{F_{n-1}\} \qquad (24)$$

and the parameters γ and β define the particular scheme to be used.

Considering again the system of decoupled equations (19) and using the relations (21) and (22) one finds that the characteristic equation for the three-time level scheme is

$$(\gamma+\beta p_i)\,\lambda^2 + [(1-2\gamma) + (1/2-2\beta+\gamma)p_i]\lambda +$$
$$[-(1-\gamma) + (1/2+\beta-\gamma)p_i] = 0 \qquad (25)$$

Writing

$$g = [1+(1/2+\gamma)p_i]/[\gamma+\beta p_i]$$
$$h = [-1+(1/2-\gamma)p_i]/[\gamma+\beta p_i]$$

the roots of (25) may be written as

$$\lambda_{1,2} = (2-g)/2 \pm [(2-g)^2-4(1-h)]^{1/2}/2$$

These roots will be complex if the quantity under the square root is negative. Then the modulus of λ is

$$|\lambda| = (1-h)^{1/2}$$

The scheme is stable if $|\lambda|<1$ for any p_i and Wood in Ref. 7 shows that this condition is satisfied if

$$\gamma>1/2 \qquad \text{and} \qquad \beta>\gamma/2$$

Equation (25) has two roots: λ_2, the principal root which is the aproximation to the exact

$$\lambda = \exp(-p_i)$$

and the spurious root λ_2. For relative stability one must have

$$|\lambda_1| < |\lambda_2|$$

Also a negative root or complex roots can produce oscillation.

In Figure 6a-f the real roots λ_1 and λ_2 and the modulus $|\lambda|$ for the complex roots are plotted against p_i for the schemes:

a) $\beta=1/3$, $\gamma=1/2$ proposed by Lees in Ref. 8

b) $\beta=3/4$, $\gamma=1$ proposed by Hogge in Ref.9

c) $\beta=0.646$, $\gamma=1.2184$ proposed by Wood [Ref. 7]

d) $\beta=4/5$, $\gamma=3/2$ proposed by Zienkiewicz [Ref. 4]

e) $\beta=9/10$, $\gamma=3/2$ proposed by the author

f) Fully implicit algorithm, $\beta=1$, $\gamma=3/2$ [Ref. 7].

From Figure 6a one may predict a very strong oscillatory behaviour for the Lees algorithm a).

In order to study the various schemes, the physical problem represented by the nondimensionalized linear heat conduction equation

$$\frac{\delta T}{\delta t} = \frac{\delta^2 T}{\delta x^2}$$

was solved in a bar of length 4 and width 1 subjected to the boundary conditions

$$T=1 \qquad at \qquad x=0$$

$$\frac{\delta T}{\delta x}=0 \qquad at \qquad x=4$$

and initial condition

$$T=0 \quad \text{at} \quad t=0$$

i.e., consider a step change in the surface temperature.
This problem was also studied by Wood and Lewis in Ref. 10.

The problem was discretized in space using ten equal
length two dimensional quadratic elements through the length
of the bar. This type of elements was chosen because it was
reported by Wood and Lewis as giving the worst numerical
results.

In order to avoid the discontinuity in the loading
term caused by the step change of the surface temperature at
the initial time, the problem was first solved starting from
time $t=\Delta t$, where Δt is the time step size. The value of the
temperature distribution at $t=\Delta t$ is provided by the exact
analytical solution. This procedure also provides the
necessary two starting vectors for the three-time level
schemes.

The temperature of the nodes at $x=1$ is used as
reference value to compare the various schemes.

In Figure 7a-d the results obtained from the
various two-time level schemes with $\Delta t=2$ for the temperature
at $x=1$ are compared with the analytical solution. For this
ideal starting conditions the Crank-Nicolson, $\theta=1/2$, proves
to be the most accurate algorithm. When the step size was
increased to 10, all the schemes performed well as shown in
Figure 8a-d.

The corresponding results for the three-time level

schemes are presented in Figure 9a-f and Figure 10a-f for $\Delta t=2$ and $\Delta t=10$, respectively.

Strong oscilations were observed with the Lees algorithm a).

For $\Delta t=2$ the schemes c), d) and e) performed well. For the larger step size of 10 the safest schemes are e) and f).

The step change in the loading term has been reported [Ref. 10] as producing relative instability when some of the schemes are used, particularly with the Crank-Nicolson algorithm. The noise may be supressed reducing the magnitude of the time step but this is impraticable in most problems. Other techniques may be used to reduce the amplitude of the oscillations as those discussed by Wood and Lewis [Ref. 10] and Gresho and Lee in Ref. 11.

The step change in the loading term in the problem in study is due to the variation of the surface temperature at the initial time $t=0$. That discontinuity in the force term is illustrated in Figure 11a. When the time dimension is discretized it has been common practice to calculate the numerical solution at the nodes $t=0$, $t=\Delta t$, etc. When a three-time level scheme is used the node at $t=-\Delta t$ is considered assuming all conditions steady before $t=0$. The force values used by this method are indicated by the circles in Figure 11a. This procedure transfers to the numerical solution the uncertainty in the value of the force

term at time t=0 introduced by the mathematical discontinuity assumed in the analytical solution. This problem is avoided if one uses the procedure illustrated in Figure 11b. The nodes considered for the discretization are those at $t=-\Delta t/2$, $t=\Delta t/2$, $t=3\Delta t/2$, etc. The discontinuity at t=0 is smoothed by the interpolation of the loading term inherent to the particular scheme being used. The values of the force term for every node are well defined and no uncertainty is associated with any of them. For the three-time level schemes the node at $t=-3\Delta t/2$ is used assuming all conditions steady before t=0.

The problem studied was solved using both starting procedures. The results obtained with the first procedure, start at t=0, using the two- and three-time level schemes are shown in Figures 12a-d and 13a-f, respectively. The second procedure, start at $t=-\Delta t/2$, gives the results shown in Figures 14a-d and 15a-f for the two- and three-time level schemes, respectively.

One may conclude that, in this simple problem, when the procedure illustrated in Figure 11b is followed, the step change in the loading term do not deteriorate the accuracy of the numerical results obtained with the schemes tested, with the exception of the Crank-Nicolson algorithm. This scheme shows an oscillatory behaviour not observed when the step change in the force term is not present.

The starting procedure of Figure 11b also proves to be an efficient way of starting the time integration with

the three-time level schemes.

2. Algorithms for Time Dependent Problems with Macro
   Program

A simplified form of the two-time level scheme is available in FEAP and is described by

$$(\frac{1}{\Delta t}[C_n] + \theta[K_n])\{T_{n+1}\} + (-\frac{1}{\Delta t}[C_n] + (1+\theta)[K_n])\{T_n\} +$$

$$\Delta t\{F\} = \{0\} \tag{26}$$

where the subscripts n and n+1 denote evaluation at time t and t+Δt, respectively.

The three-time level scheme programmed in FEAP is described by the expression

$$(\gamma[C_n] + \beta \Delta t[K_n])\{T_{n+1}\} + ((1-2\gamma)[C_n] + (1/2-2\beta+\gamma)\Delta t[K_n])\{T_n\}$$

$$+ (-(1-\gamma)[C_n] + (1/2+\beta-\gamma)\Delta t[K_n])\{T_{n-1}\}$$

$$+ \{F_n\} = \{0\} \tag{27}$$

where the subscripts n, n+1 and n-1 denote evaluation at times t, Δt+ t and t-Δt respectively.

It must be noted that in (26) and (27) the vector $\{F_n\}$ is an aproximation of the interpolated force vector $\{\bar{F}\}$.

If the force vector is strongly dependent on time, this aproximation may produce wrong results.

In problems where the stiffness matrix is constant and the force term results from the specified boundary displacements or where the specified boundary forces are only time dependent, the interpolation of the force term may be easily acomplished since the force/displacement boudary values can be changed at any time using the macro commands MESH or PROP. This procedure was found partcularly useful

38

when a step change in boundary displacements or forces is applied at the initial time.

The first order ordinary differential equation solver module is acessed using the macro command ODE1 followed by one of the following macro commands:

INIT to specify the initial displacement vector
and the integration constants

LINE to perform the two-time level algorithm

QUAD to perform the three-time level algorithm

Since the matrices [K] and [C] and the vector {F} must be formed before the time integration, the macro commands TANG, CMAS or LMAS and FORM are closely associated with the use of ODE1. It should be noted that the macro FORM forms the vector

$$\{R(T_n)\} = -\{F_n\} - [K_n] \, \{T_n\}$$

which is allways dependent on the current displacement. Therefore the macro command FORM must be used every time step and preceed ODE1.

To solve a fully nonlinear problem the following macro program may be used:

```
DT              Δt
ODE1 INIT
LOOP            n
TANG
CMAS ( or LMAS )
FORM
ODE1 LINE ( or QUAD )
```

39

```
TIME

DISP

NEXT
```

The macro command program must be followed by the data cards necessary to specify the integration constants θ, γ and β and the initial displacement vector.

If the matrices [K] and/or [C] are not temperature or time dependent then the instructions TANG and/or CMAS or LMAS must be placed outside the loop.

For the simplest case of a linear problem, the macro program may be

```
DT              Δt

ODE1 INIT

TANG

CMAS ( or LMAS )

LOOP        n

FORM

ODE1 LINE ( or QUAD )

TIME

DISP

NEXT
```

V. NUMERICAL EXAMPLES

A. RADIAL TEMPERATURE DISTRIBUTION IN A HOLLOW
    CYLINDER

Consider the hollow cylinder of infinite length already
discussed in section III. Consider the case in which the
inside wall is maintained at a constant temperature and heat
is lost by convection through the outer wall. This problem
was considered for solution by Lew [Ref. 3].

The geometrical and thermophysical characteristics are:

Inside radius (r_{in})	0.1 m
Outside radius (r_{out})	0.3 m
Thermal conductivity (k)	10 W/m °C
Outside ambient temperature	20 °C
Heat transfer coefficient	200 W/m^2°C
Inside wall temperature (T_{in})	820 °C

The finite element model of this problem was completly
discussed before and is shown in Figure 1. The complete
computer output is given in Appendix B. It must be noted
that the element arc width used is arbitrary since the
problem is symmetric and no heat is conducted perpendicular
to the radius.

The comparision between the analytical and FEAP
solutions is shown in Figure 15. In this case the values
provided by the finite element model used are very accurate
and no further mesh refinement is necessary.

B. NONLINEAR STEADY STATE HEAT CONDUCTION IN A SLAB
   WITH TEMPERATURE DEPENDENT CONDUCTIVITY

A liquid is boiled by a flat electric heater plate of thickness 2L. The internal energy Q generated electrically may be assumed to be uniform. The boiling temperature of the liquid, corresponding to a specific pressure, is T_w.

In the finite element analysis for the temperature distribuition in the heater, ten equal length quadratic elements were used to represent the unit cross section through the half plate thickness L, since the problem is symmetric. The extremity of the resulting slab is assumed at the temperature T_w while the other is insulated.

The thermal conductivity is assumed temperature dependent according to the expression

$$k = a(1+bT)$$

where a and b are constants.

Assuming the following values for the parameters in a consistent system of units

   $T_w = 0.0$
   $a = 0.5$
   $L = 1.0$
   $Q = 1.0$

the problem was solved for b equal 0.0, 1.0 and 2.0.

The numerical results are compared with the analytical solutions in Figure 16 and one may conclude that FEAP provides exact solutions in this problem.

C. TRANSIENT TEMPERATURE DISTRIBUITION IN A HOLLOW CYLINDER

The same hollow cylinder of infinite length treated before is considered again but the case solved is the one with both outside and inside walls maintained at constant temperatures. Thermal diffusivity, $\alpha = k/\rho c$ is specified as 0.00005 m^2/sec. Initially the cylinder is at an ambient temperature T_0 of 20°C; suddenly the temperature of the inside wall is raised to 820°C while the outside wall is maintained at 20°C. All other conditions are the same as for the steady state problem.

In order to obtain the unsteady solution the number of elements in the finite element model was doubled, thus twelve radial quadratic elements were used. The element arc width chosen is the same 6° as before.

The two time-level scheme with $\theta = 2/3$ was used for the time integration. In order to compare the results the dimensionless temperature T*, radius r* and Fourier number Fo were used. They are defined as

$$T^* = (T - T_0)/(T_{in} - T_0)$$
$$r^* = r/r_{in}$$

and

$$Fo = \alpha t/r_{in}^2$$

A time step size Δt of 2 seconds was chosen initially. After 10 sec, Δt was increased to 10 sec and to 50 sec after 200sec. After 1000 sec, Δt was increased to 100 sec.

Figure 17 shows the evolution of the temperature of the

43

center interior points of the cylinder and Figure 18 compares the analytical values with the FEAP solution for the radial temperature distribuition for two different times.

D. SLAB WITH RADIATION-CONVECTION BOUNDARY CONDITIONS

Consider a plane slab of thickness L, with constant thermophysical properties (k= ρ c=1), subjected to simultaneous radiation and convection at x=0 and insulated at x=L (0<x<L). The slab is initially at a uniform temperature T_0 and it was decided to consider zero ambient temperatures for convection and radiation.

It was also decided to consider the Biot number

$$Bi = h_c L/k = 1$$

where h_c is the convection heat tranfer coefficient and

$$R = \epsilon \sigma T_0^3 L/k = 4$$

where ϵ is the emissivity and σ the Stefan-Boltzman constant.

The same mesh as in Example B was employed for the finite element solution. The time integration was performed by the three-time level scheme with γ =1/2 and β=1/3. A variable time step size was chosen. A value of Δt=0.001 was used initially. After t=0.02, Δt was increased to 0.0025. After t=0.2, Δt was increased to 0.01. After t=0.4, Δt was increased to 0.025. After t=1.0, Δt was increased to 0.05.

The history of the ratios T_w/T_0 and T_m/T_0 are plotted in Figure 19 against the Fourier number

$$Fo = \alpha t/L^2$$

T_w is the surface tmperature at x=0 and T_m is the surface temperature at x=L.

The finite element results are compared with those given by Haji-Sheick and Sparrow in Ref. 12 and obtained from a Monte Carlo method.

VI. CONCLUSIONS AND RECOMMENDATIONS

The computer program in this thesis provides an accurate and reliable means for solving a variety of Heat Transfer problems. The use of this program and efforts to increase its versatility are highly encouraged.

The capabilities of the program, while quite significant, can still be improved. The present version is designed for an "in-core" solution technique, which restricts the problem size to within the computer core-size. The capacity to handle large problems may be increased through the use of some "out of core" technique for solving the system of equations and even for building the mesh data. In that case the size of the problems treated would be restricted only by the availability of external storage devices.

In order to check the mesh input and to easily interpret the output, a graphics module must be included in the program.

Although the time integration module seems to provide reliable results in linear and some nonlinear problems, it should be subjected to further research in order to test it with different types of nonlinear analysis.
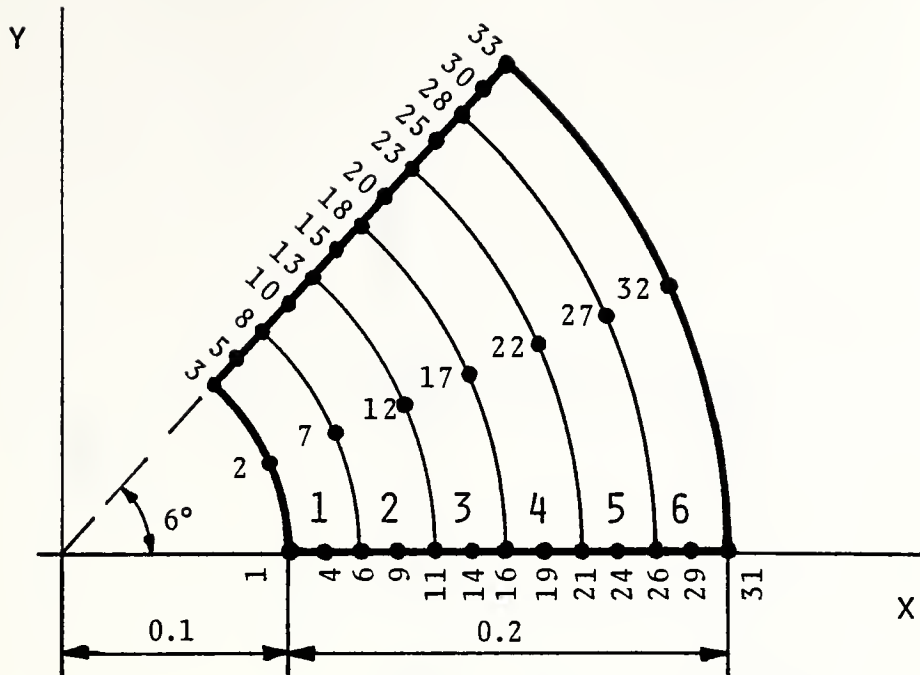
Figure 1. Mesh for Hollow Cylinder Problem
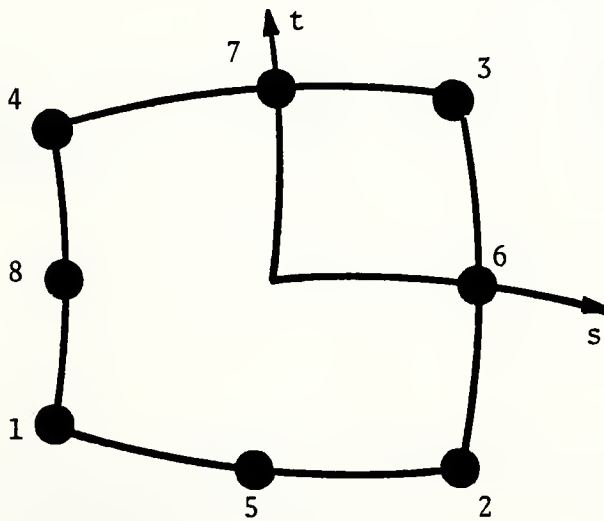
Figure 2. Eight-Node Serendipity Quadrilateral

47
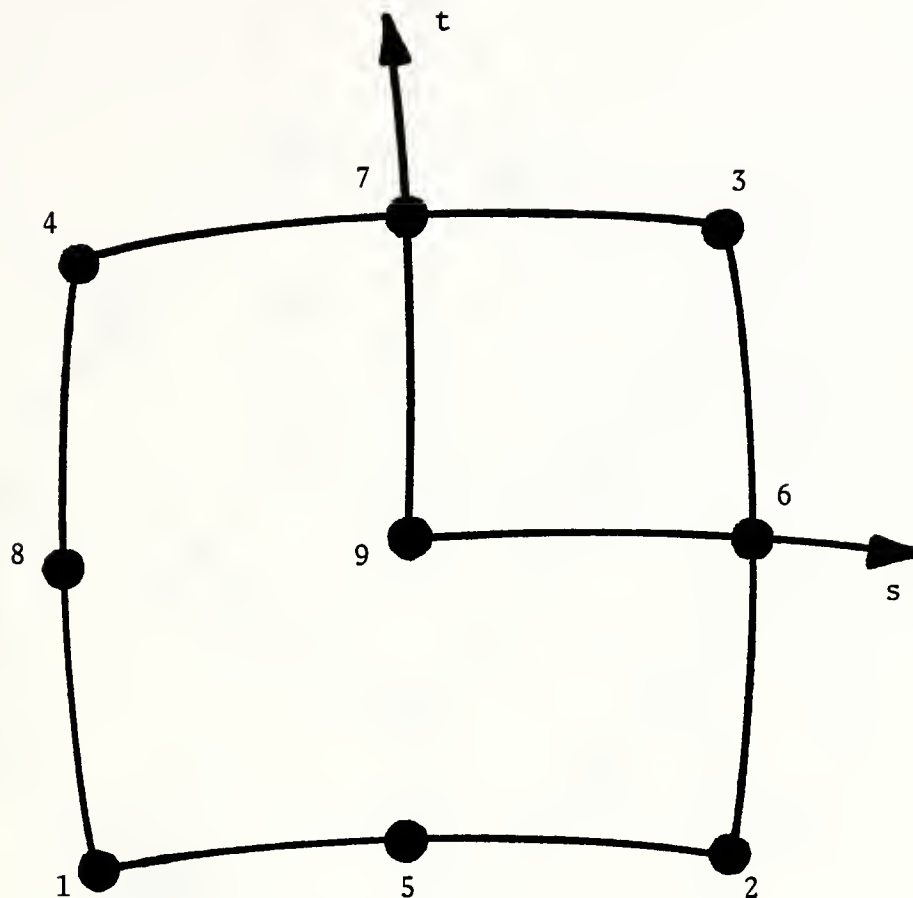
Figure 3.   Local Node Number Sequence for a Quadratic
            Lagrangian Element
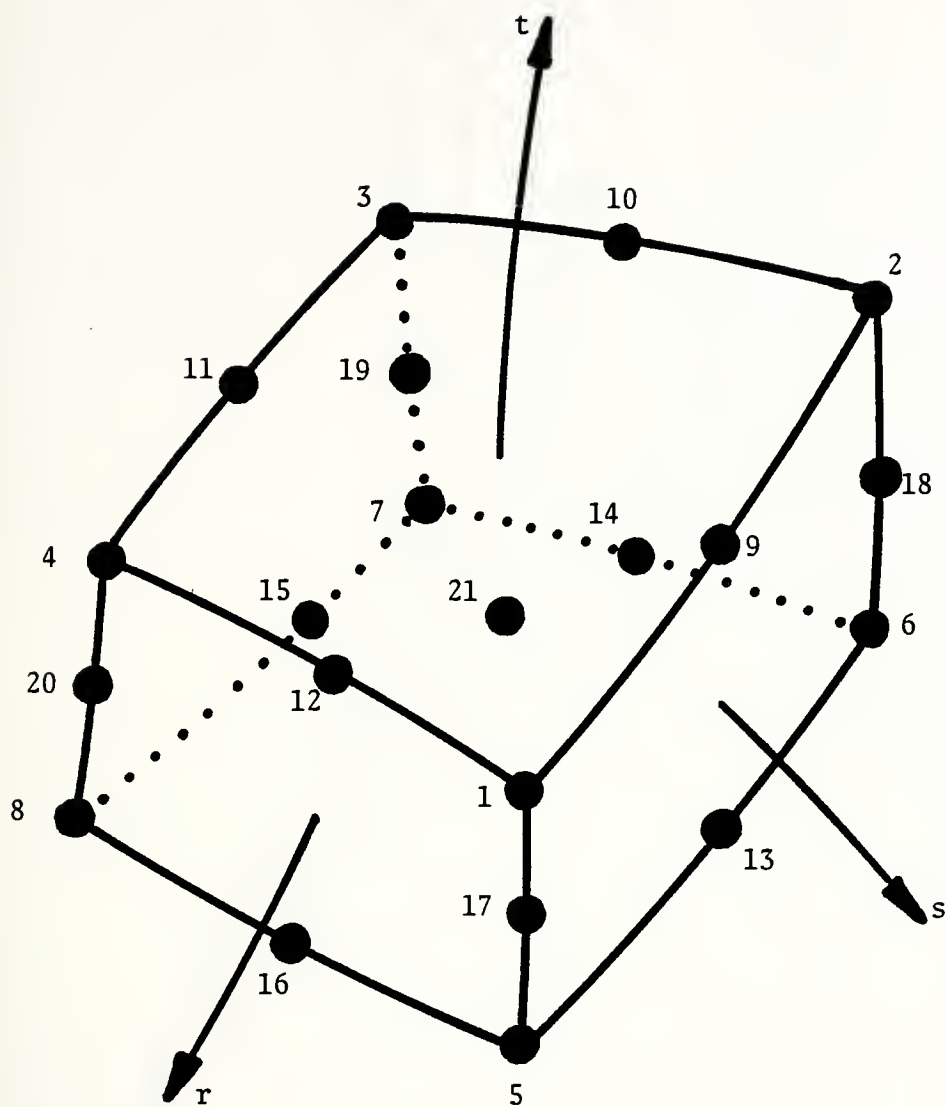
Figure 4. Local Node Number Sequence for a 21-Node
            Three Dimensional Element

49
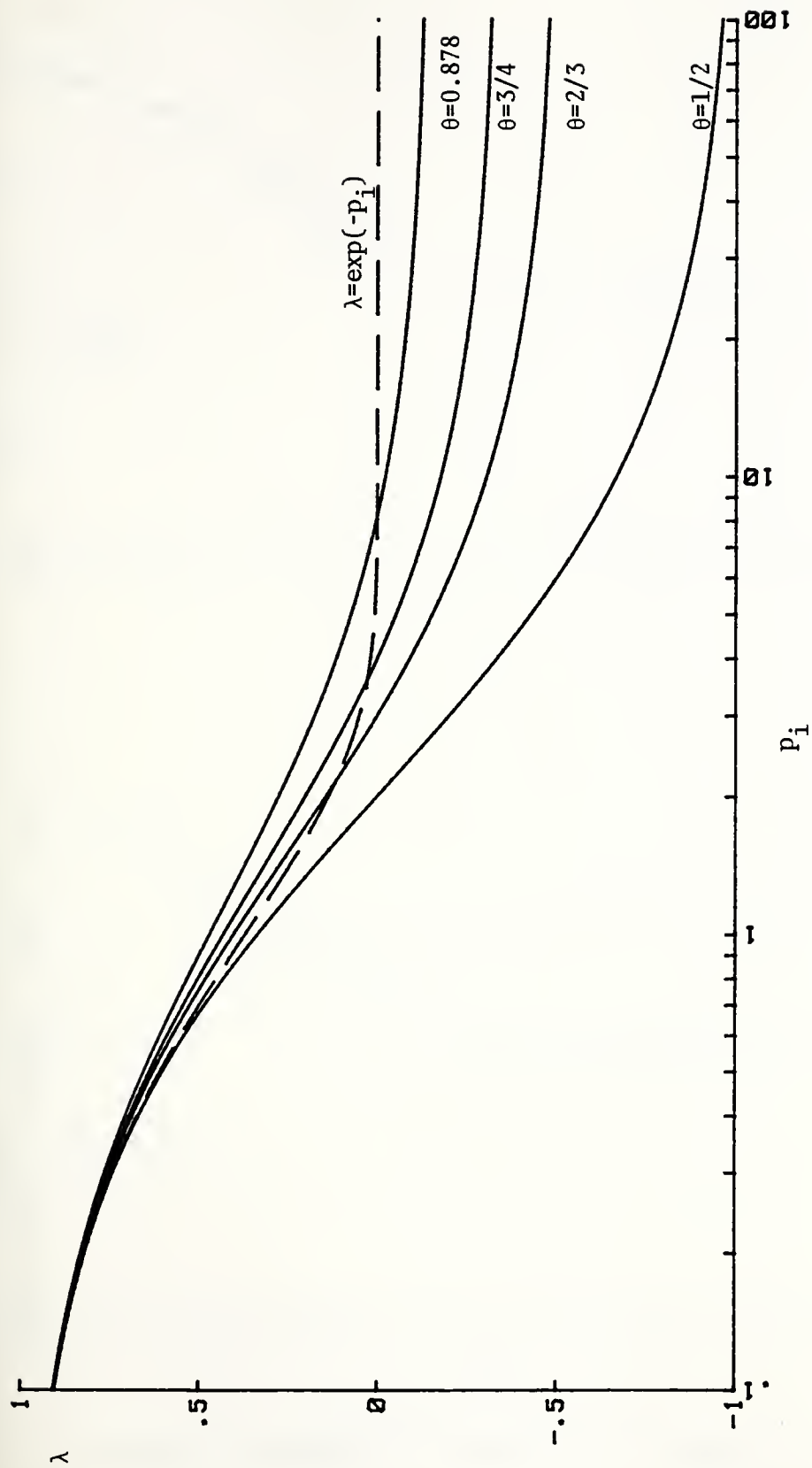
Figure 5. Zienkiewicz Two-Time Level Scheme
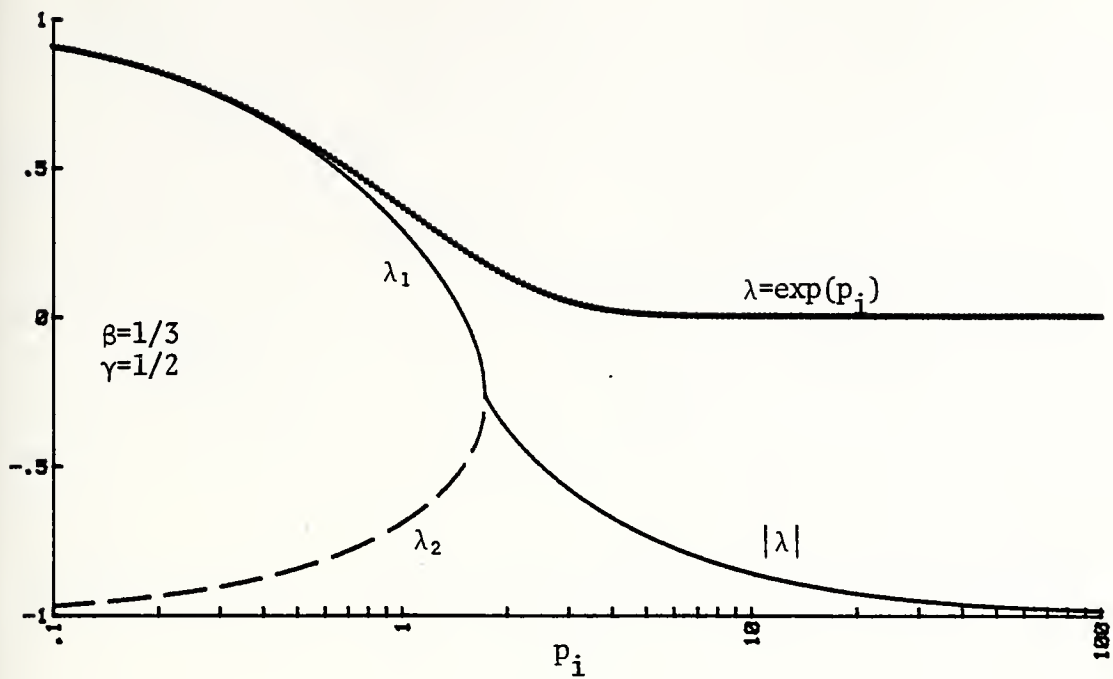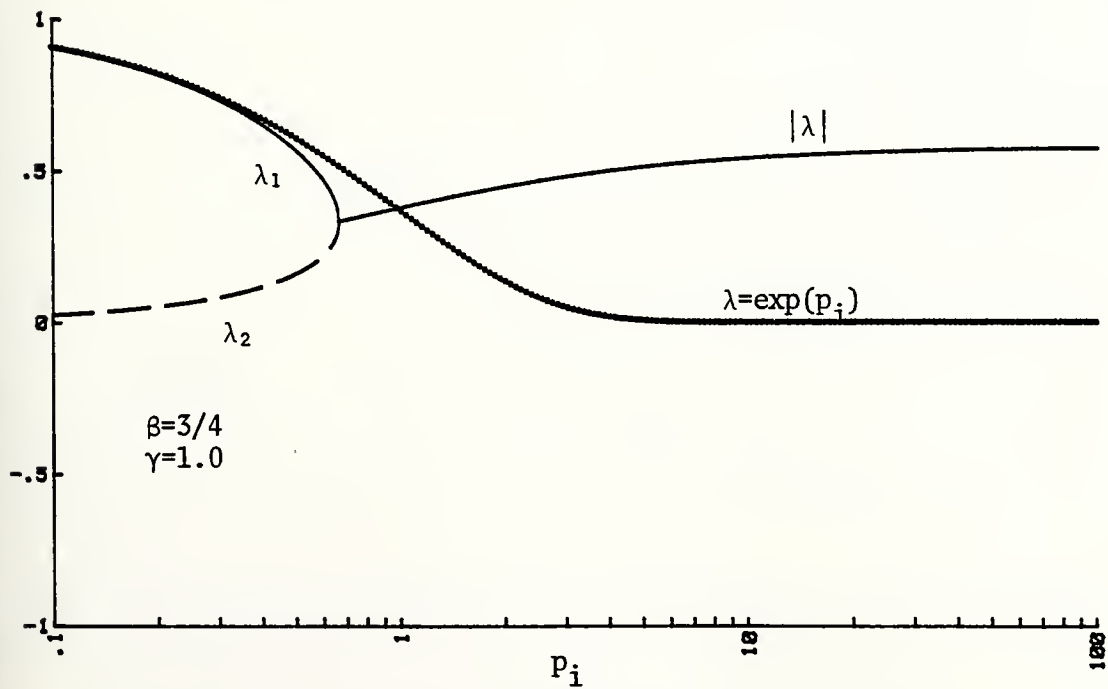
Figure 6.   Zienkiewicz Three-Time Level Schemes

51
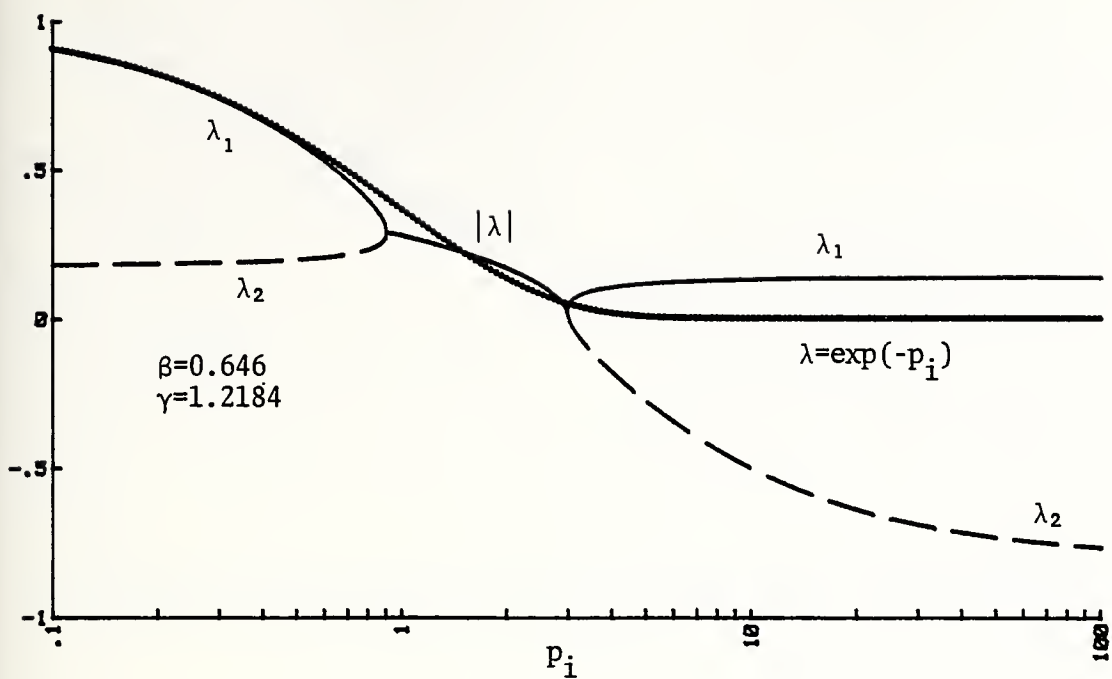
(c)
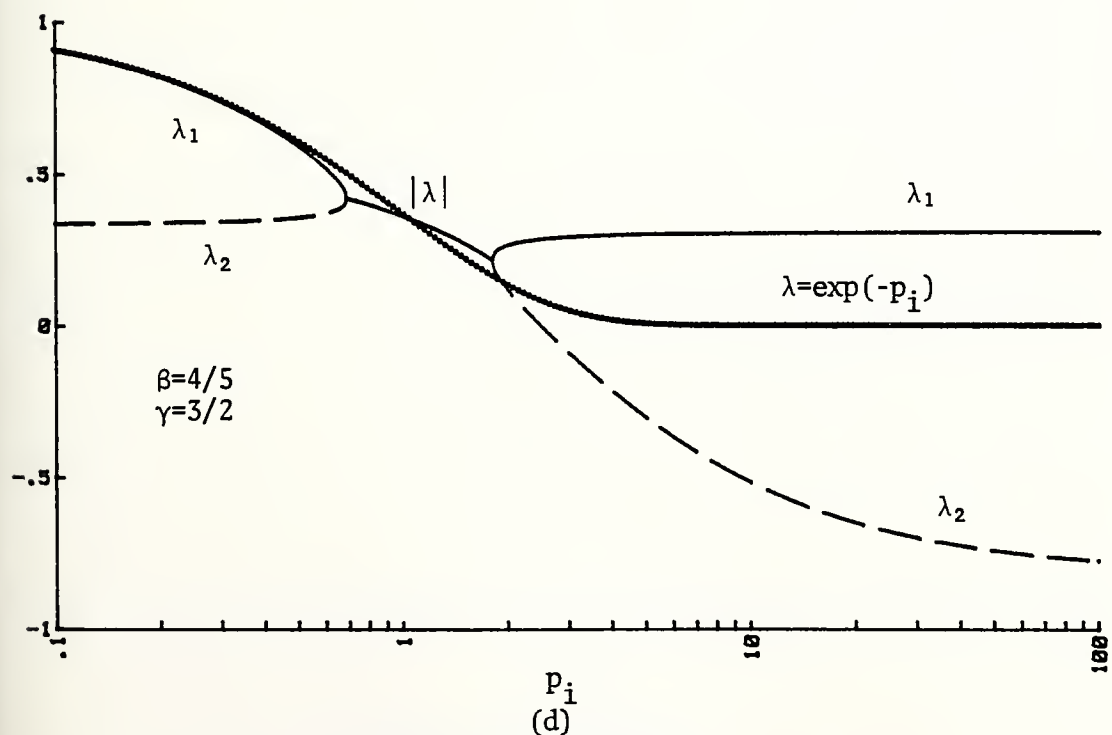
(d)

Figure 6. (continued)

52
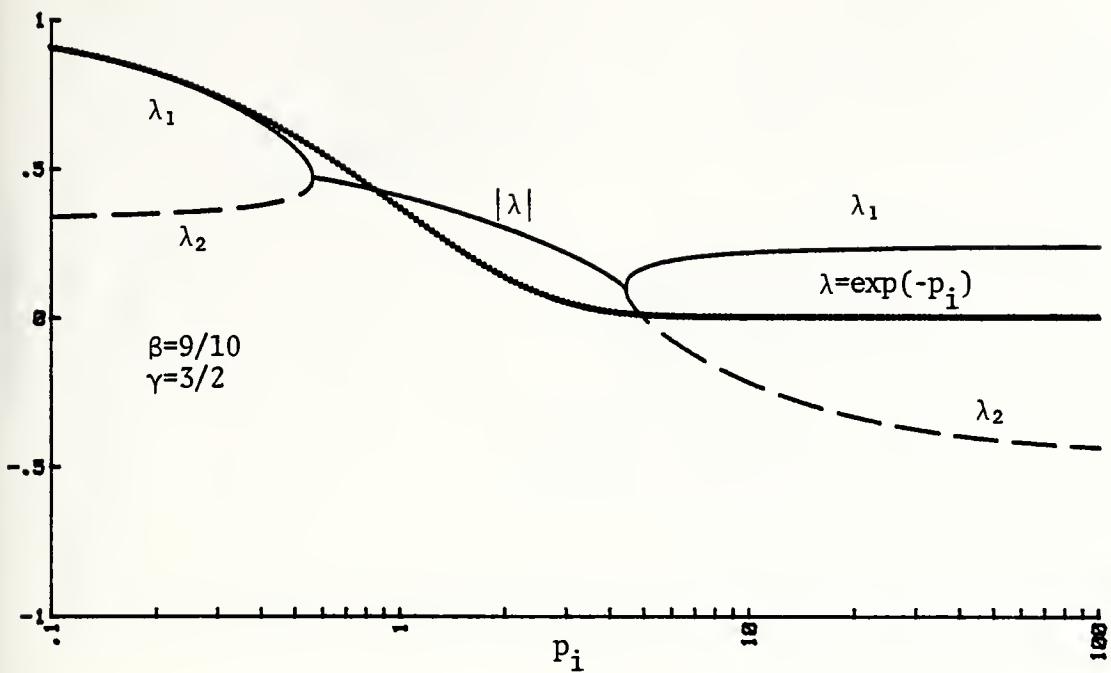
(e)
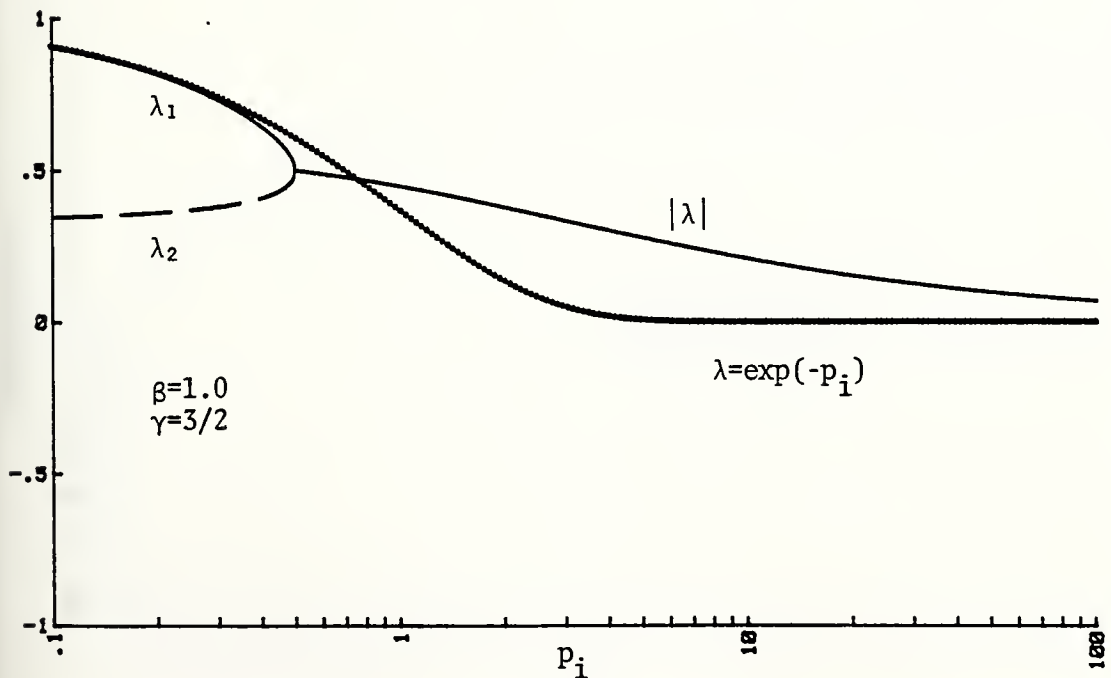
(f)

Figure 6. (continued)

53

(a)

(b)

Figure 7.

54
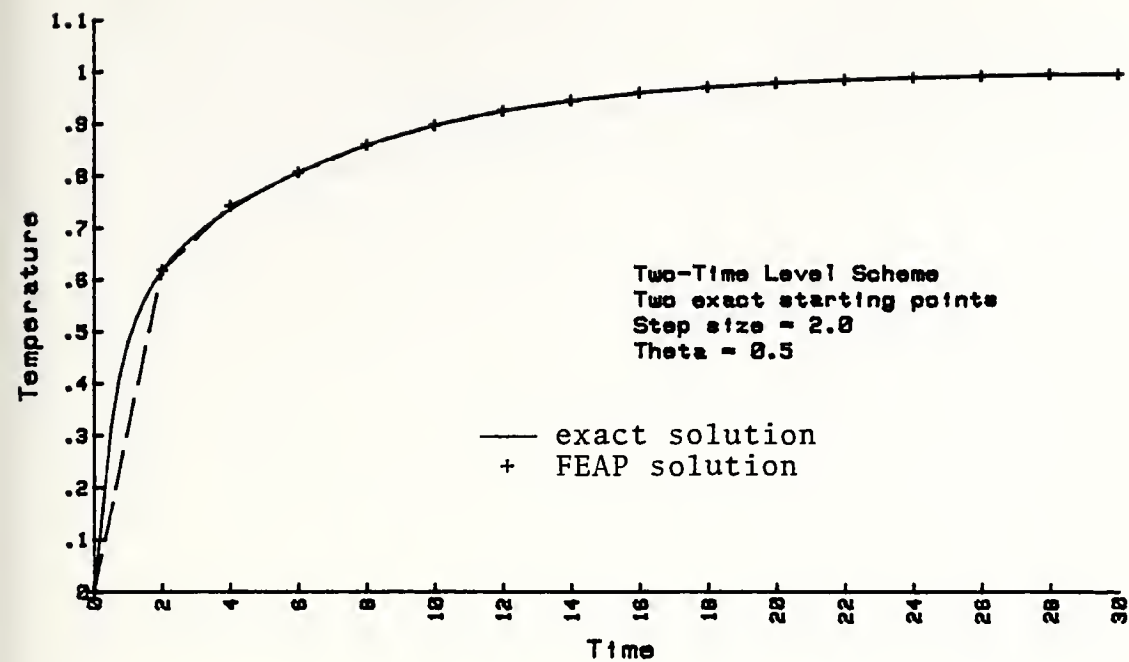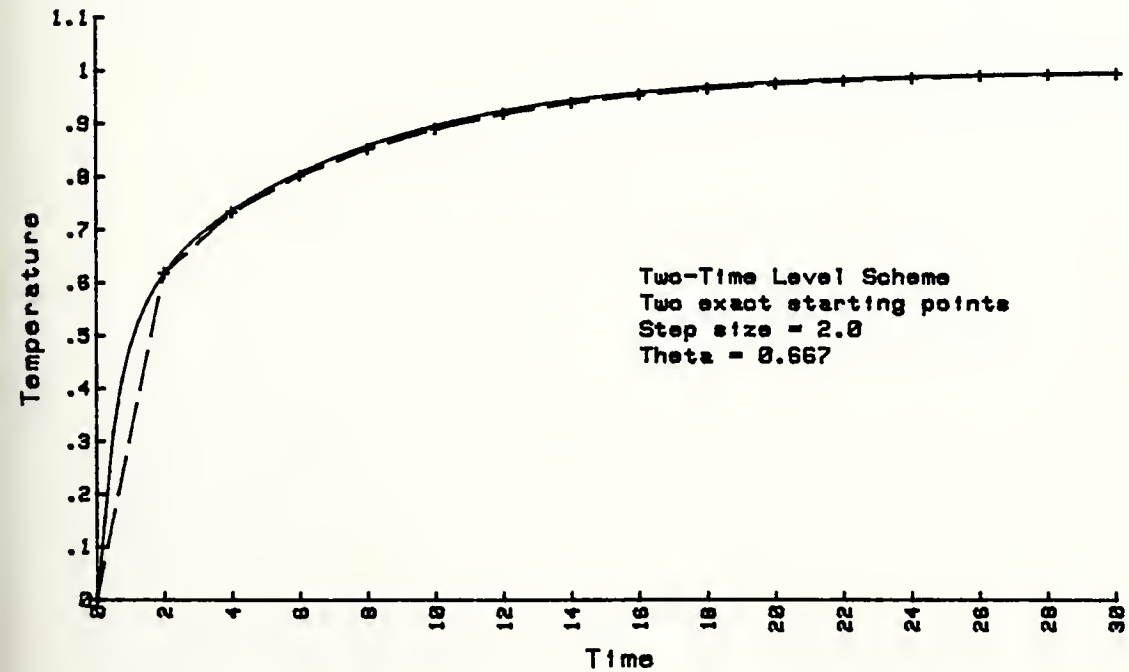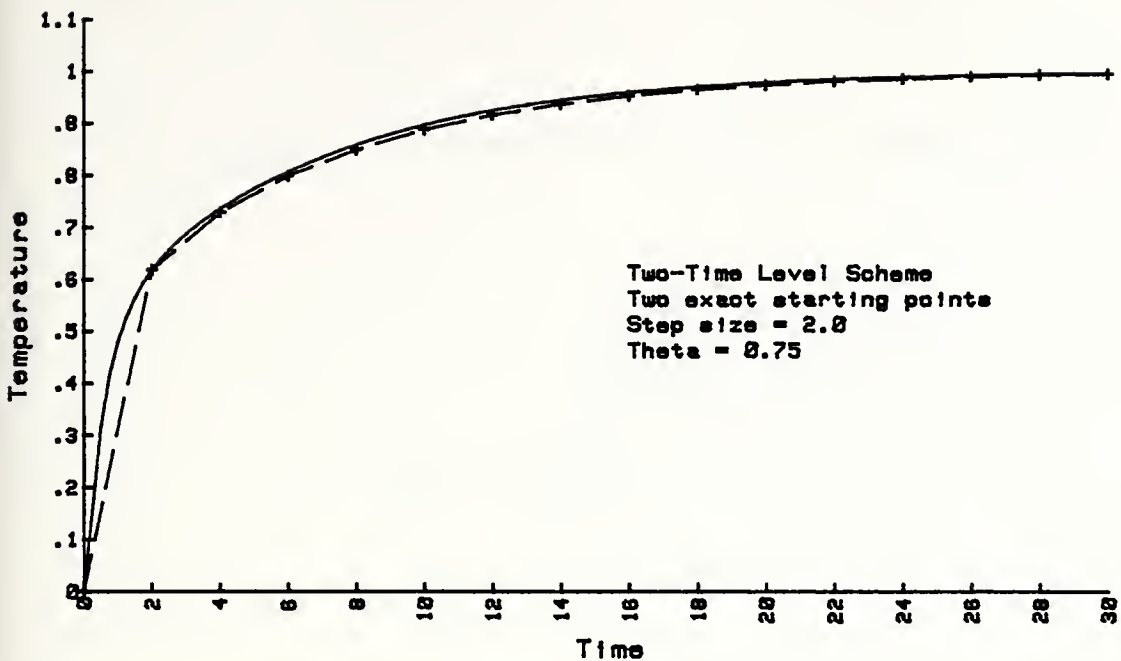
(c)
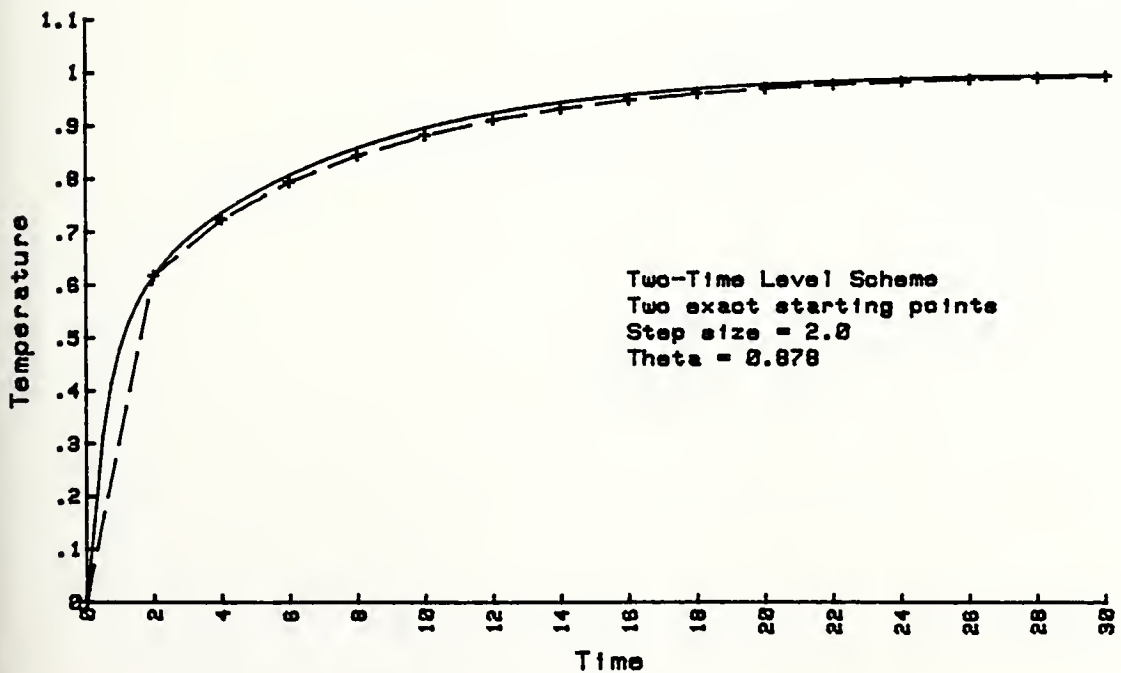
(d)

Figure 7. (continued)

55
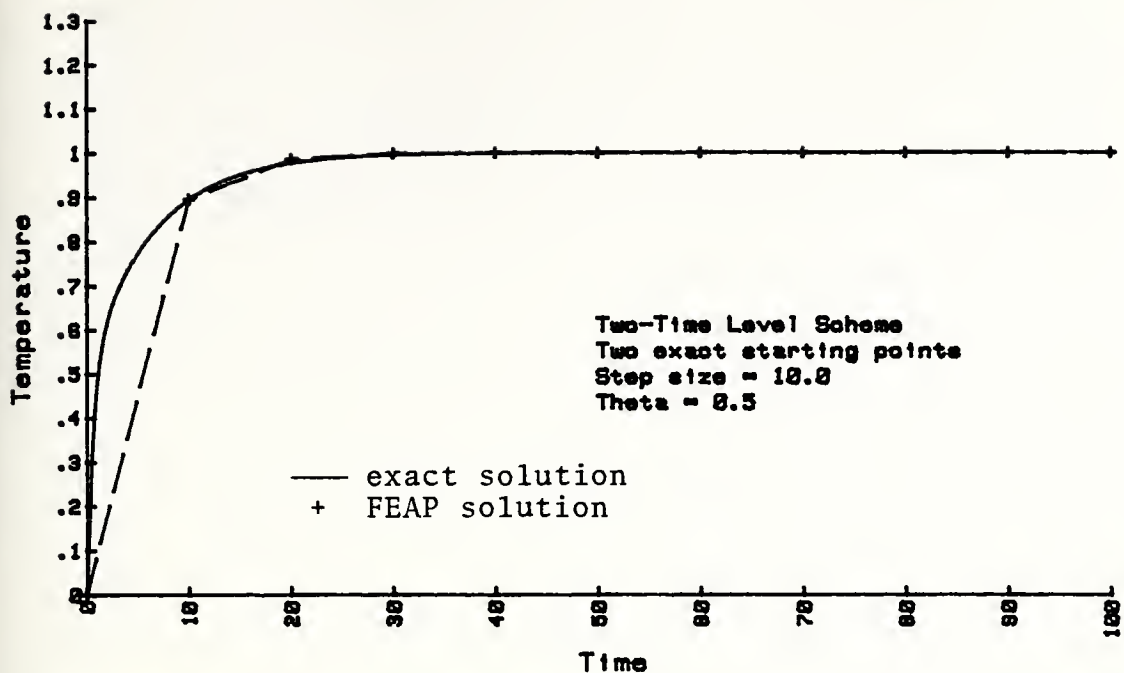
(a)
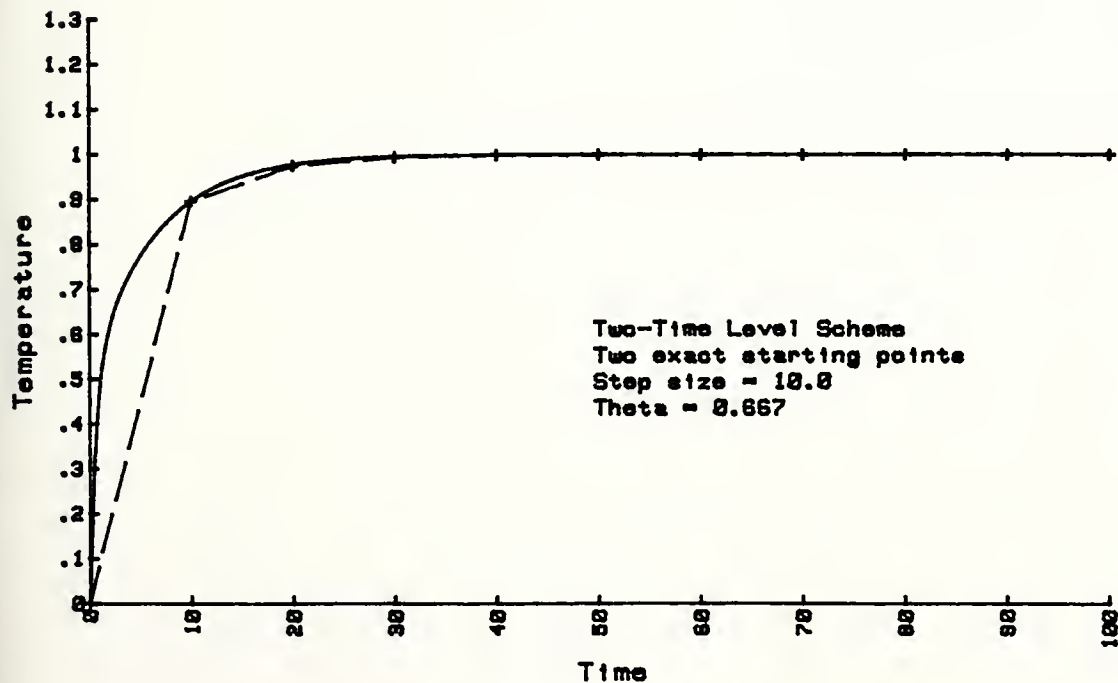
(b)
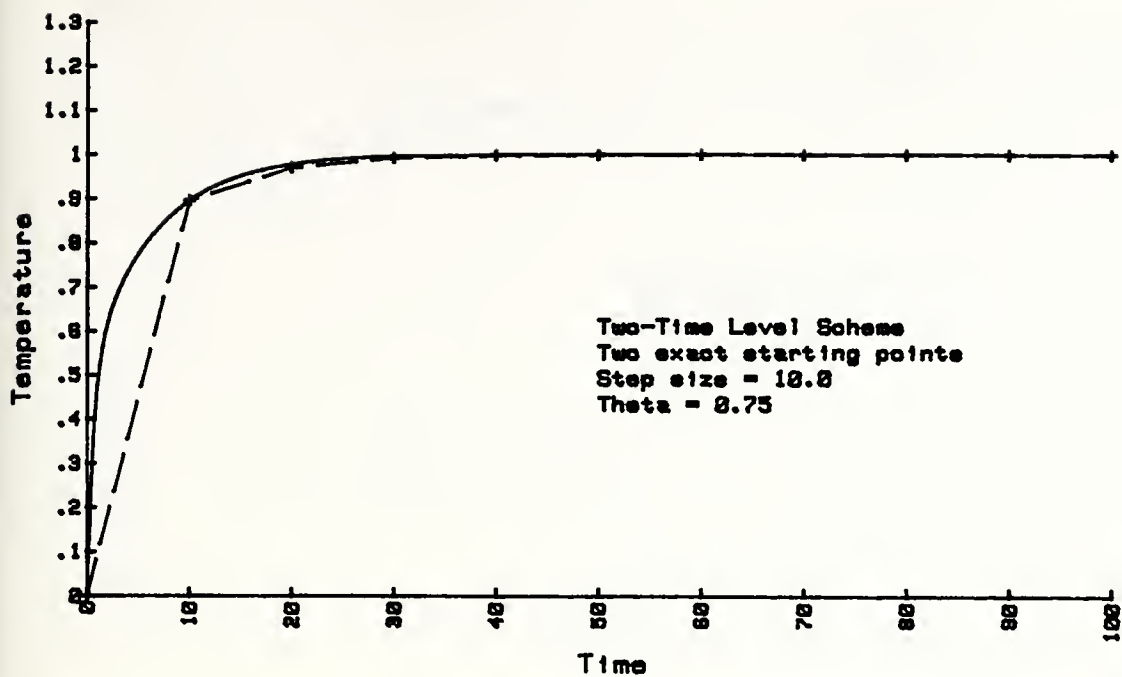
Figure 8.

(c)
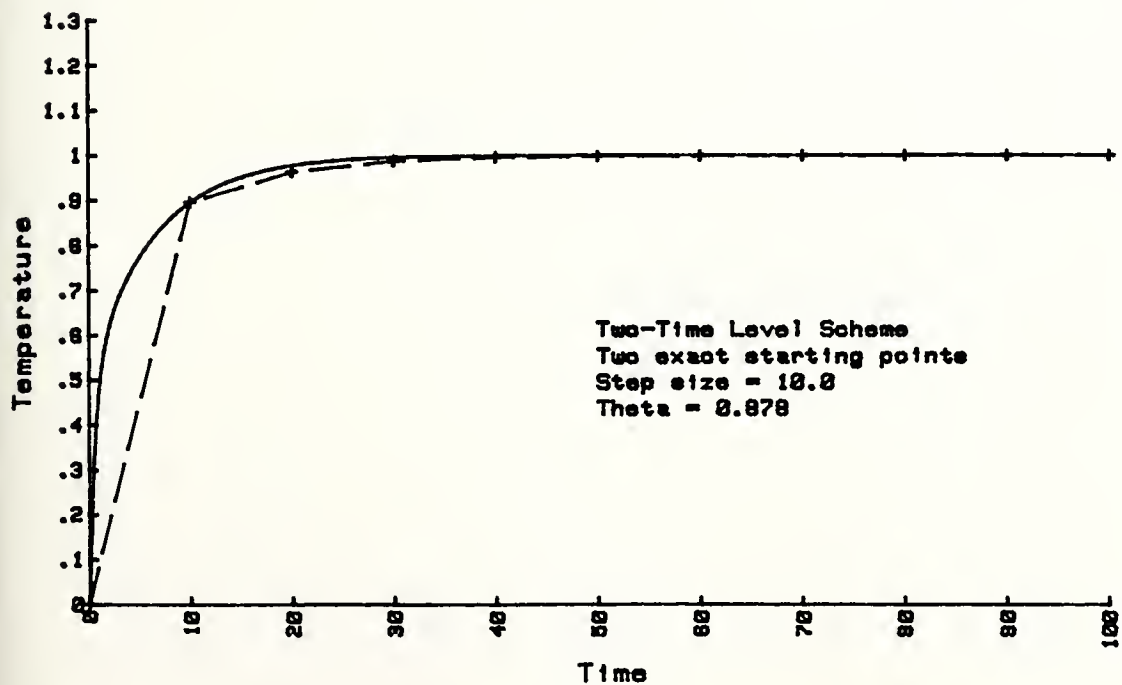
(d)

Figure 8. (continued)

57
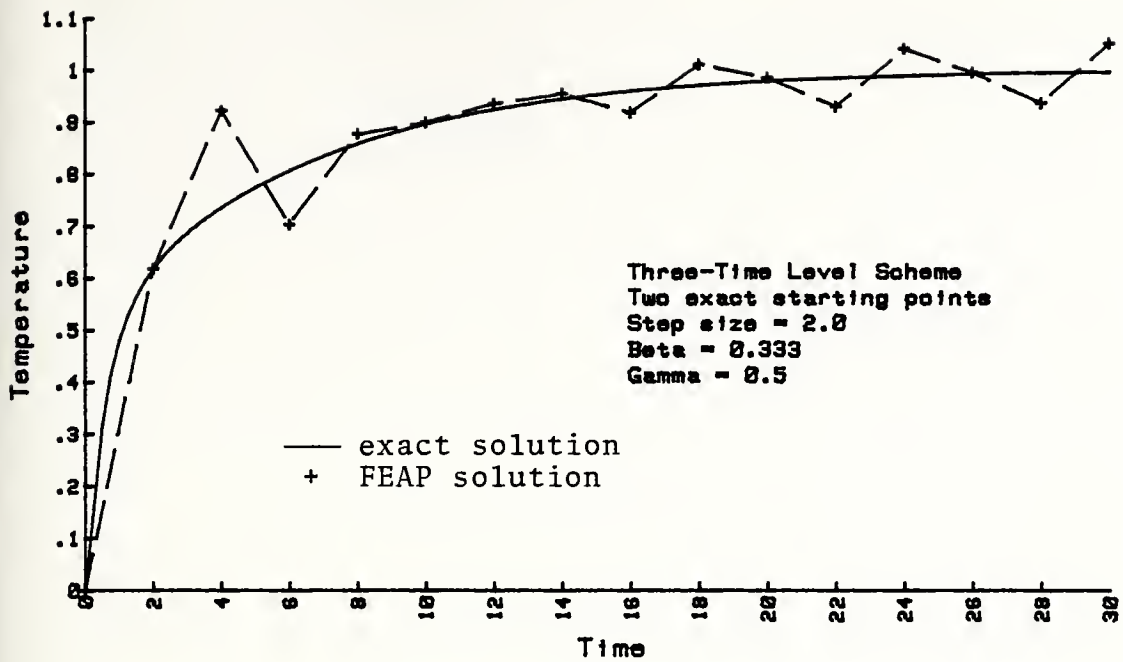
(a)
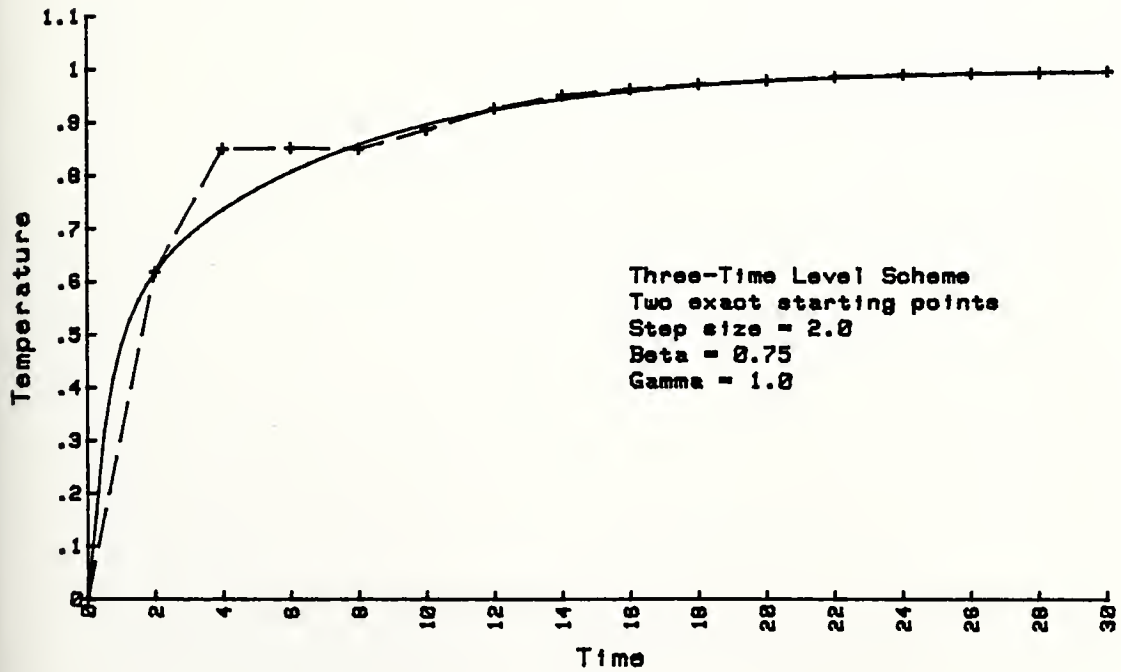
(b)
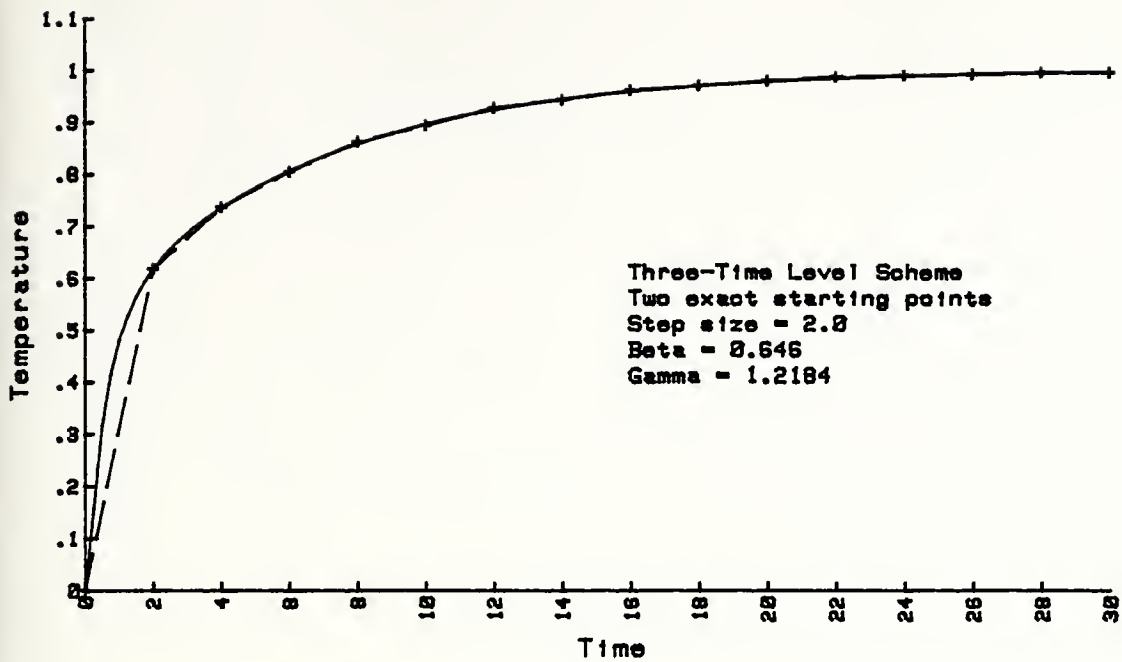
Figure 9.

(c)
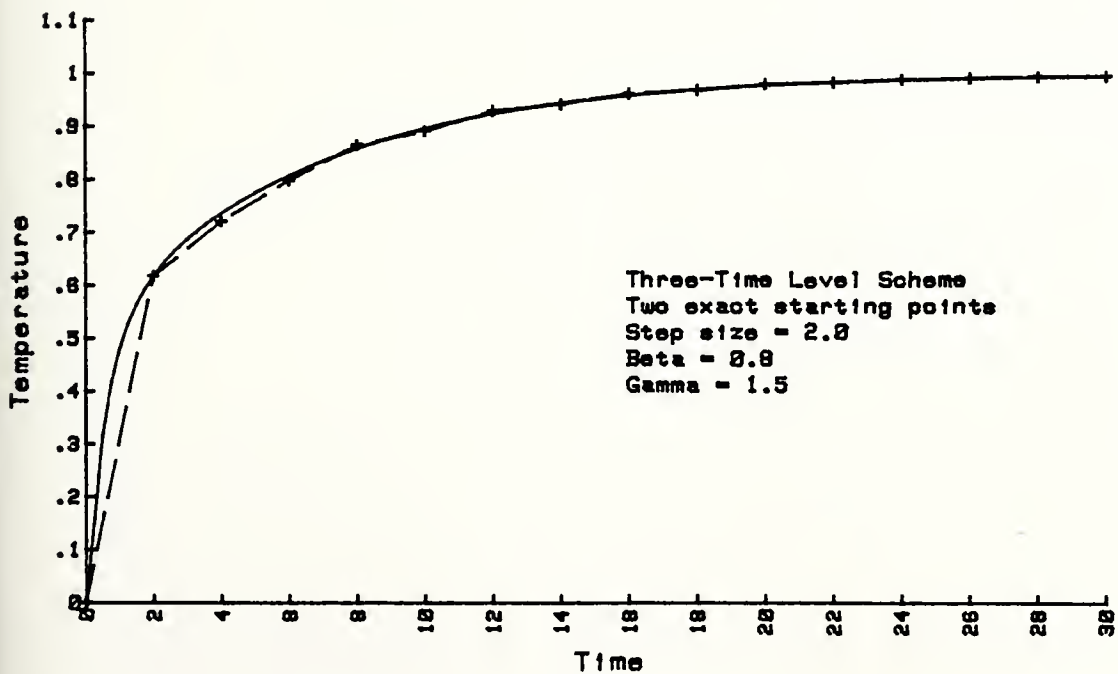
(d)

Figure 9. (continued)

59
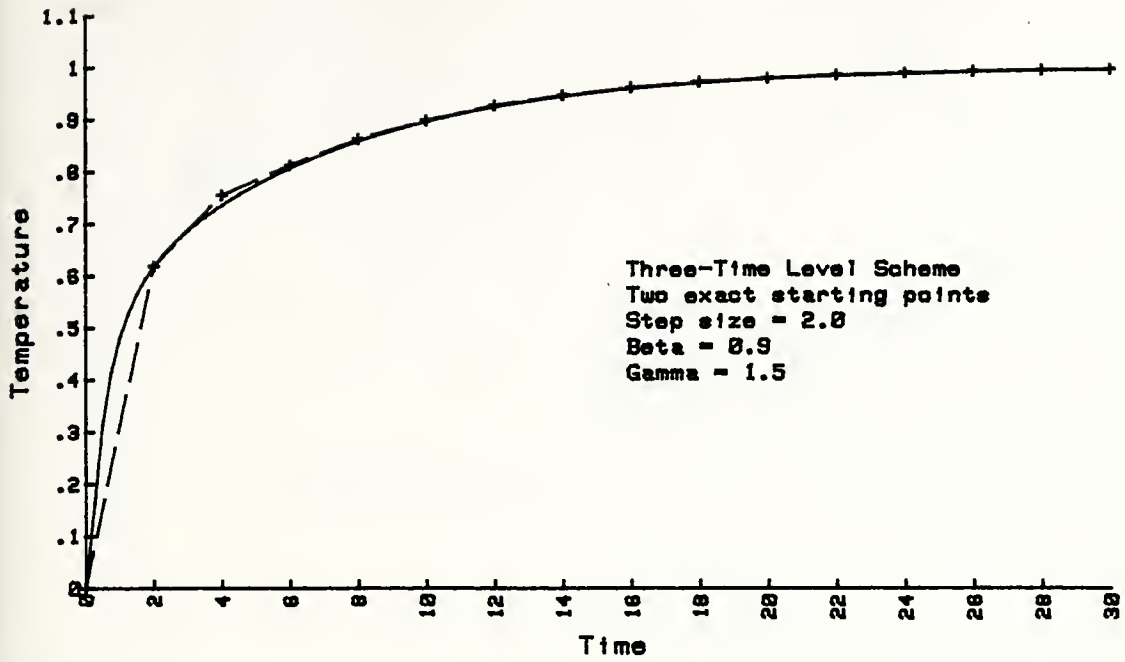
(e)
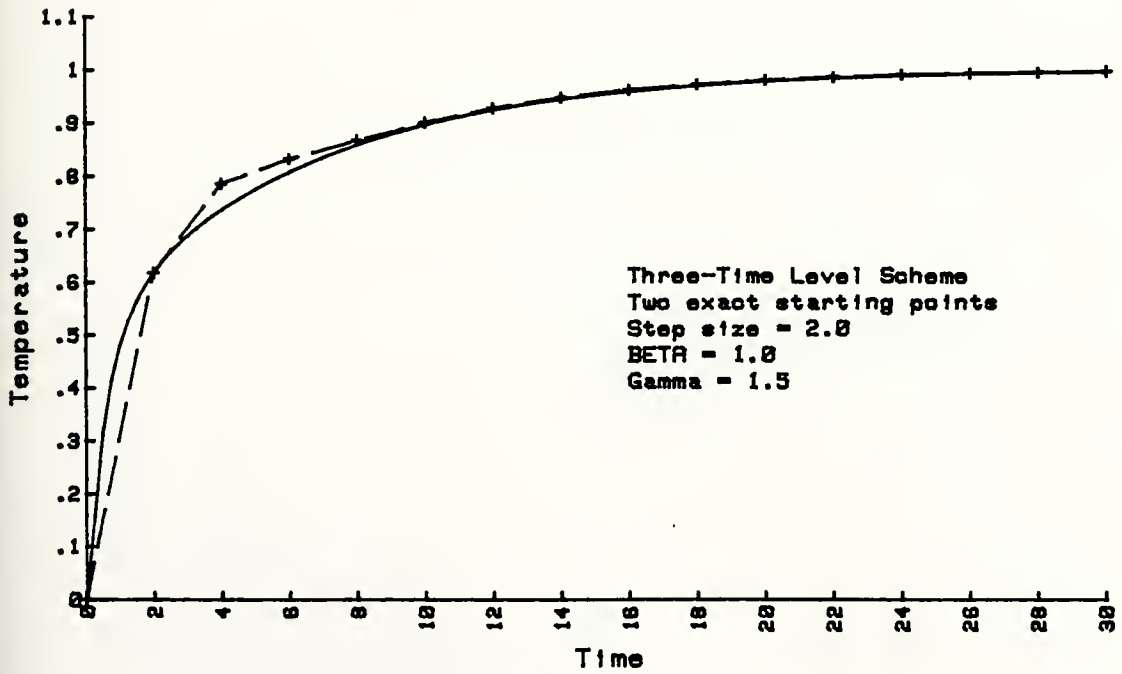
(f)

Figure 9. (continued)

60
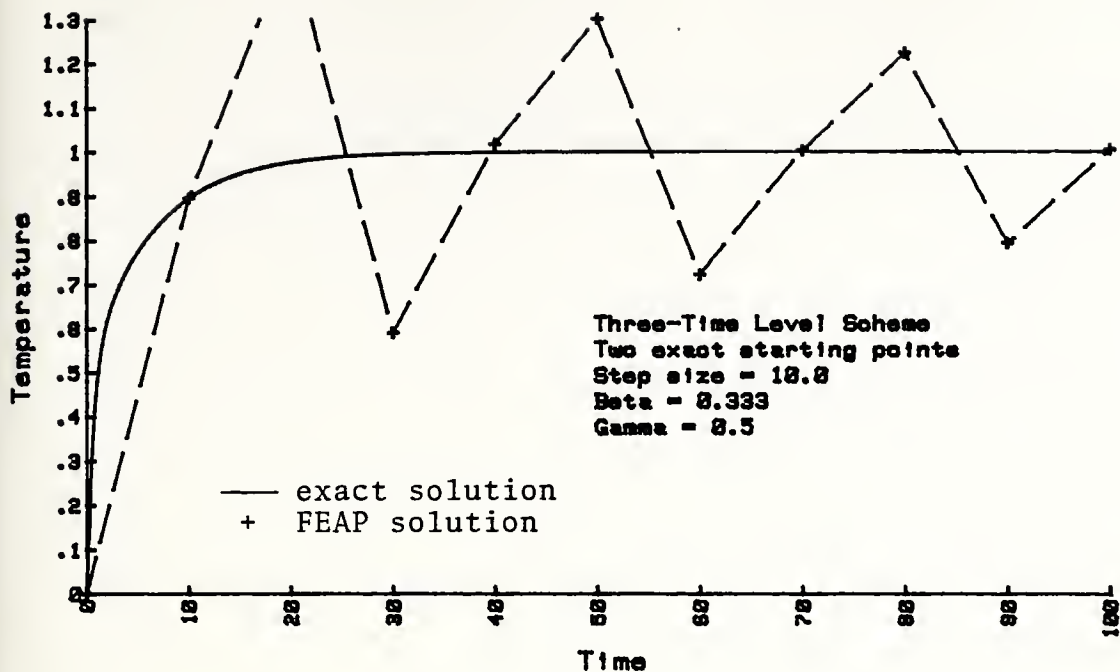
(a)
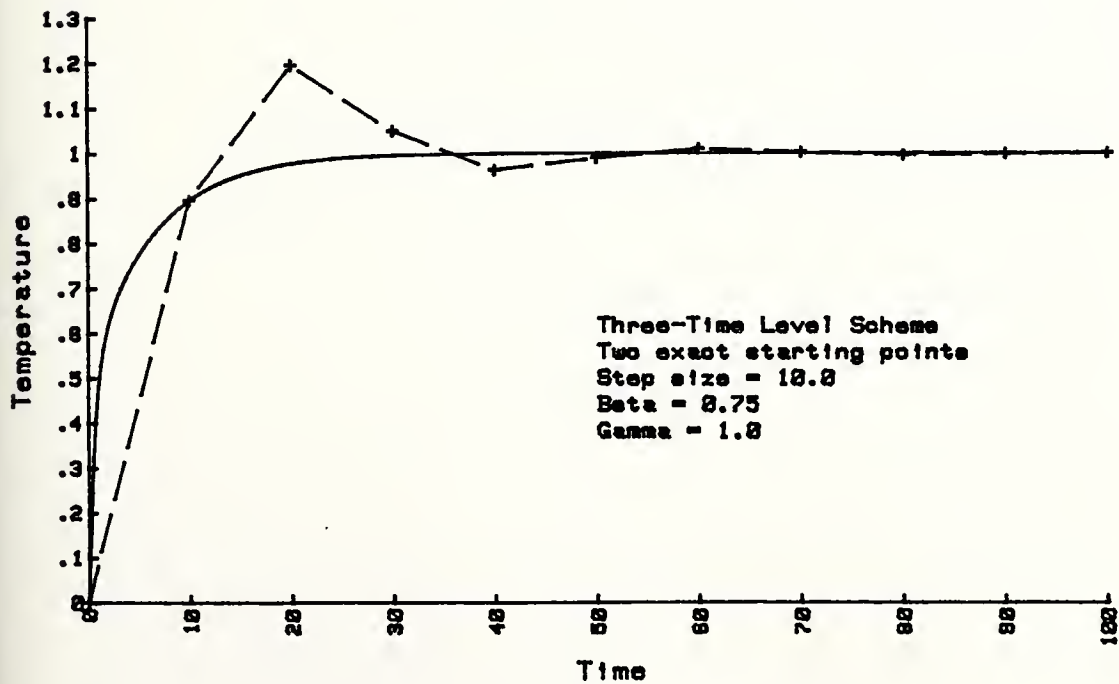
(b)

Figure 10.

(c)

(d)

Figure 10. (continued)

62

Three-Time Level Scheme
Two exact starting points
Step size = 10.0
Beta = 0.9
Gamma = 1.5

(e)

Three-Time Level Scheme
Two exact starting points
Step size = 10.0
Beta = 1.0
Gamma = 1.5
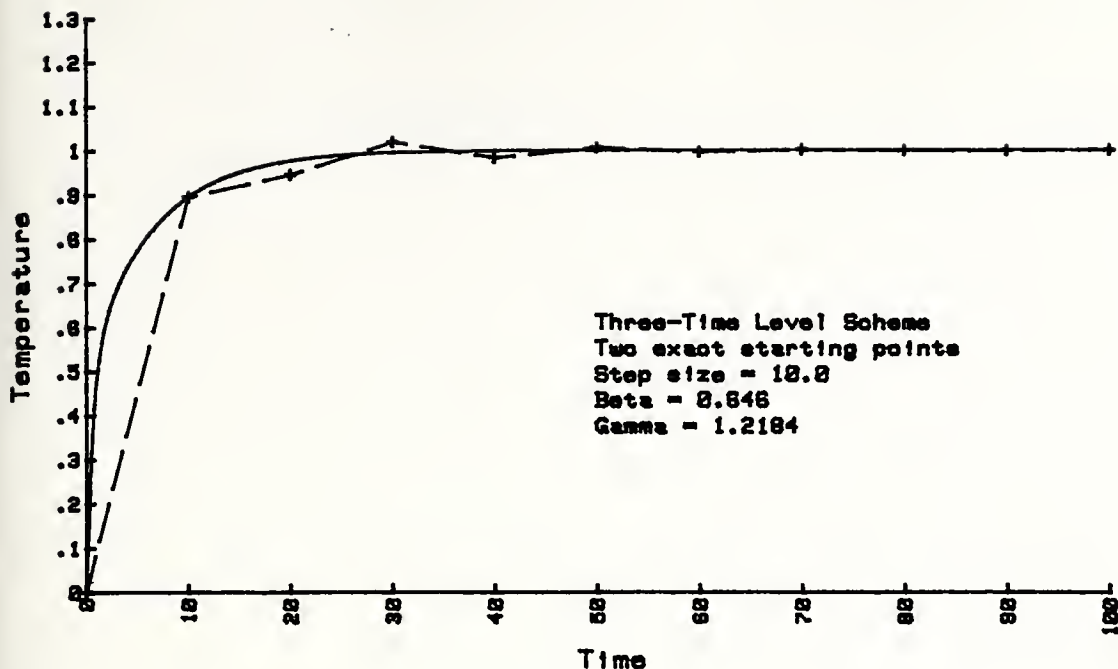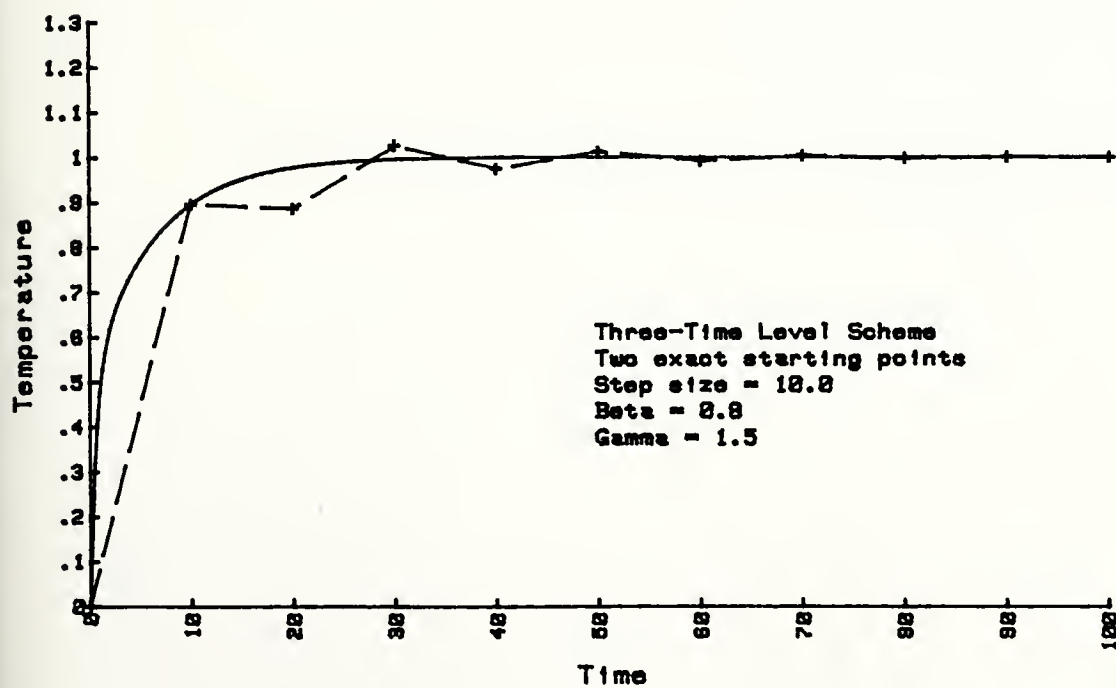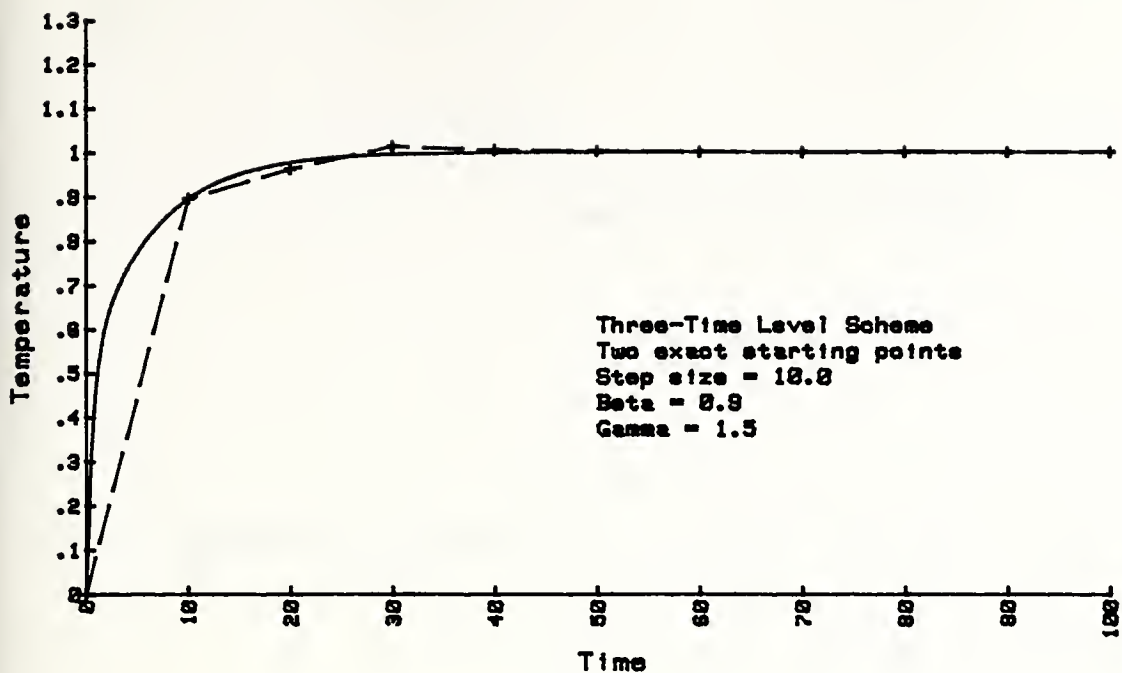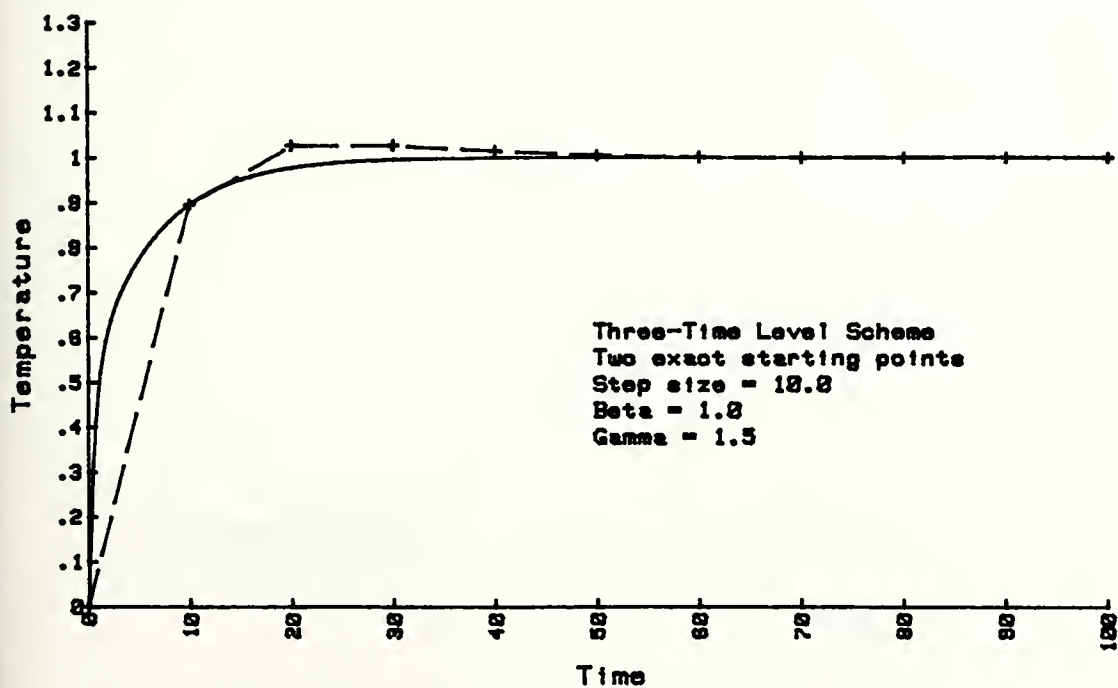
(f)

Figure 10. (continued)

63
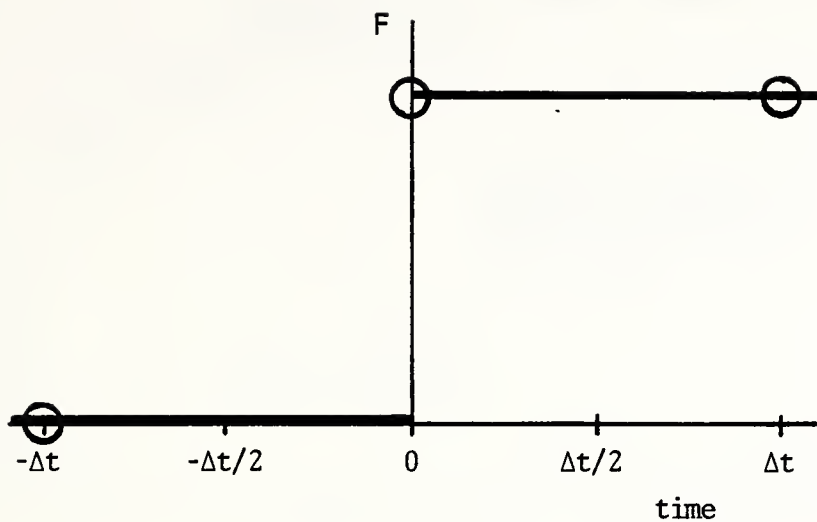
(a)
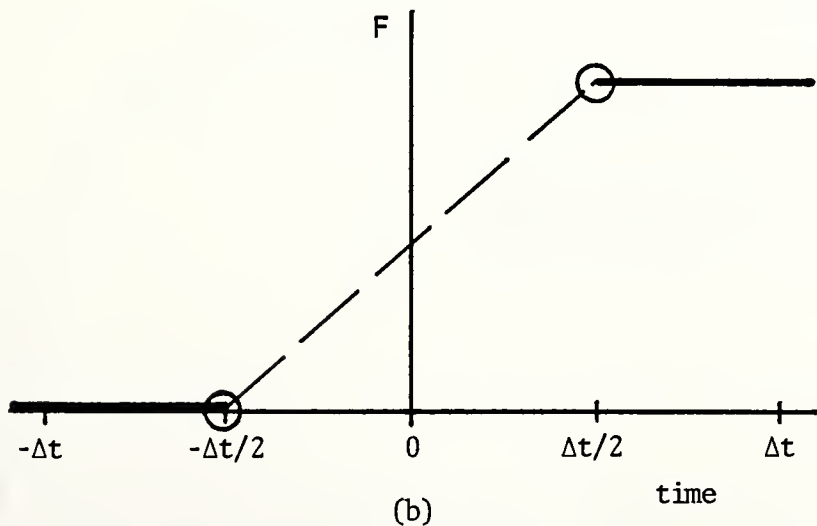
(b)

Figure 11. Step Change in the Loading Term

(a)

(b)

Figure 12.

Two-Time Level Scheme
Step size = 2.0
Start at time = 0.0
Theta = 0.75

(c)

Two-Time Level Scheme
Step size = 2.0
Start at time = 0.0
Theta = 0.878
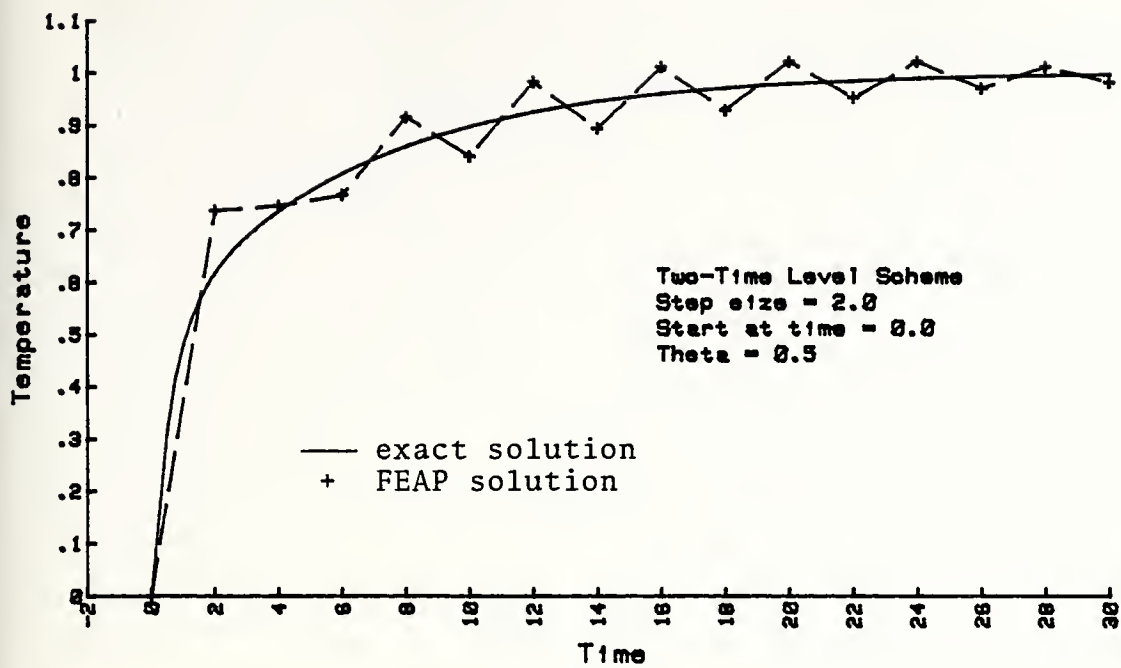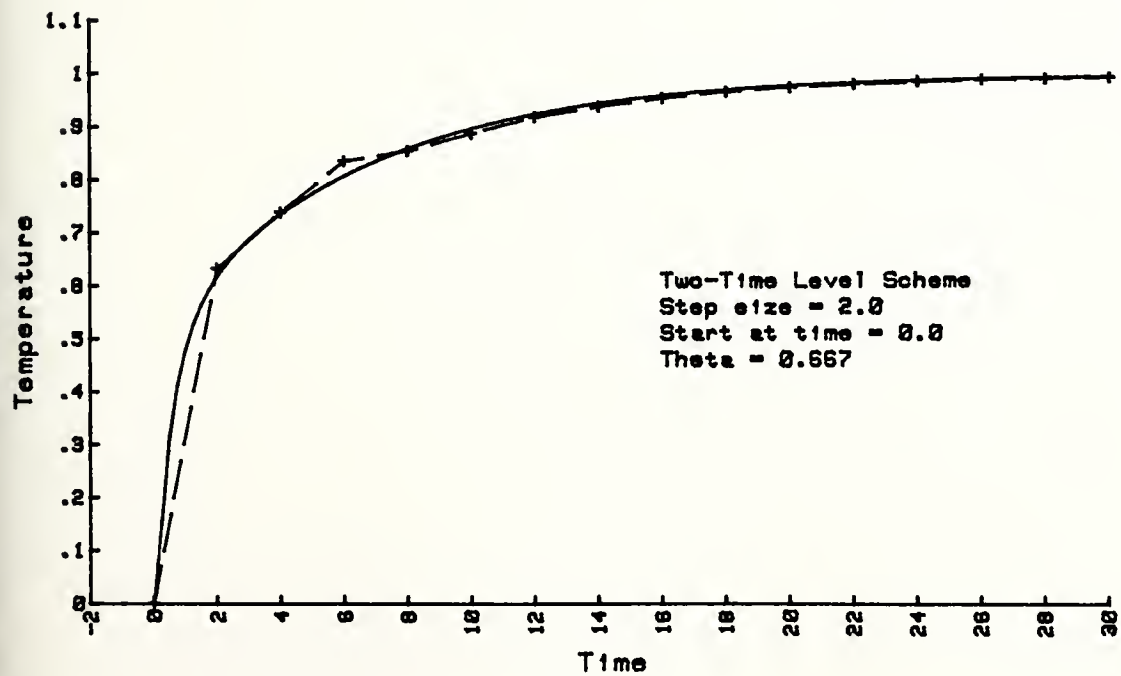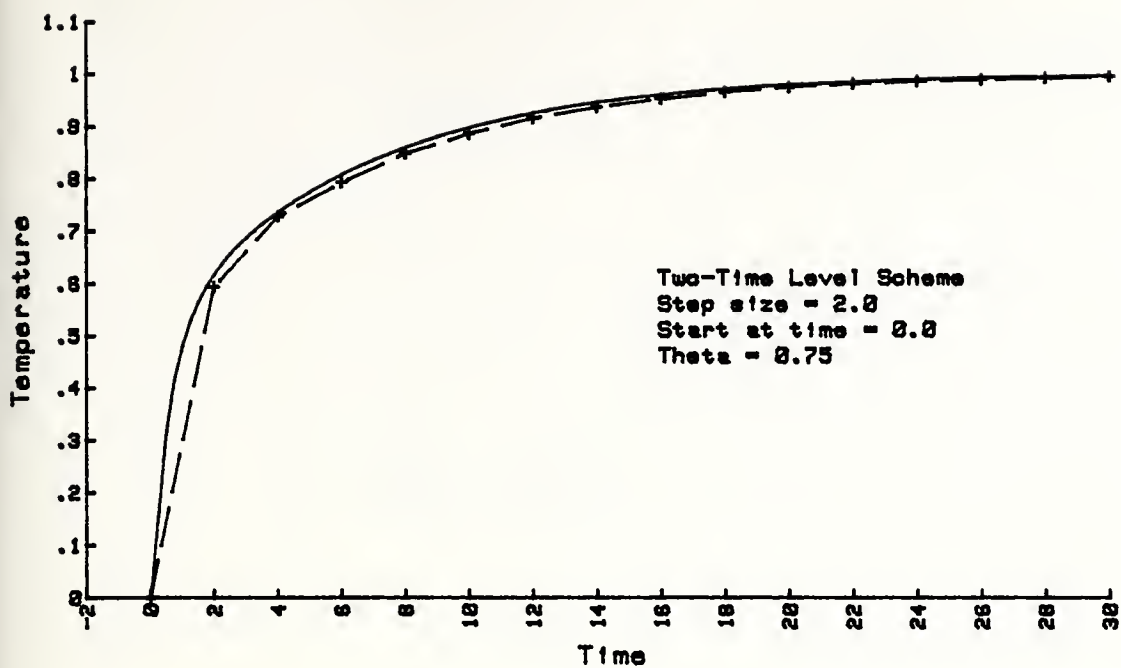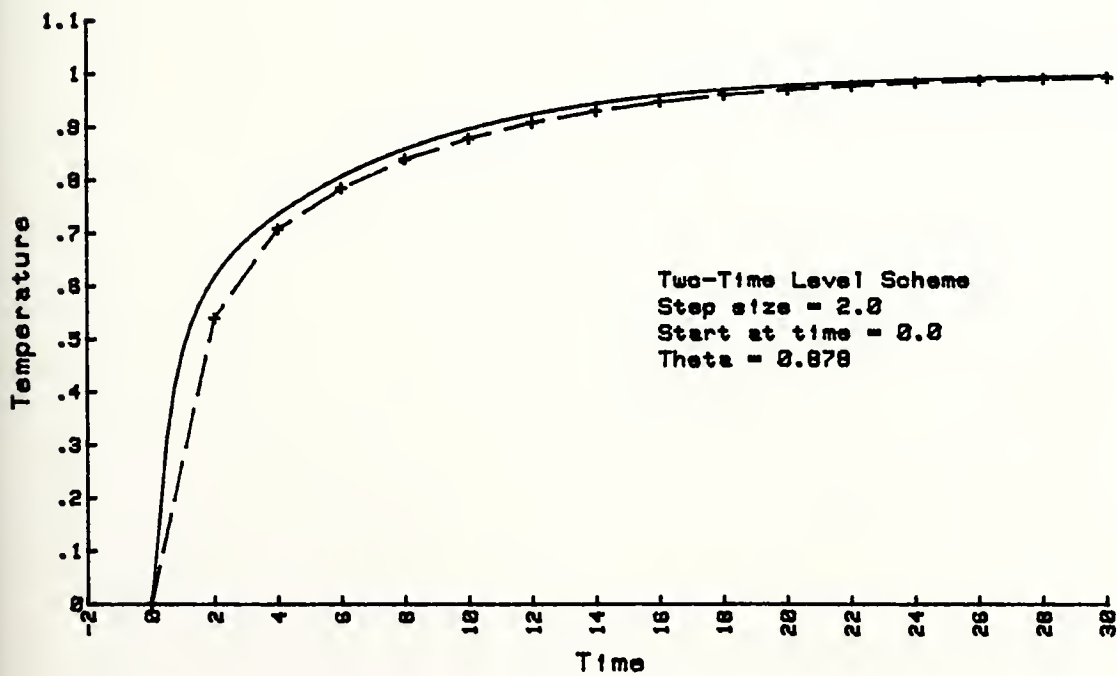
(d)

Figure 12. (continued)

66

(a)

(b)

Figure 13.

67
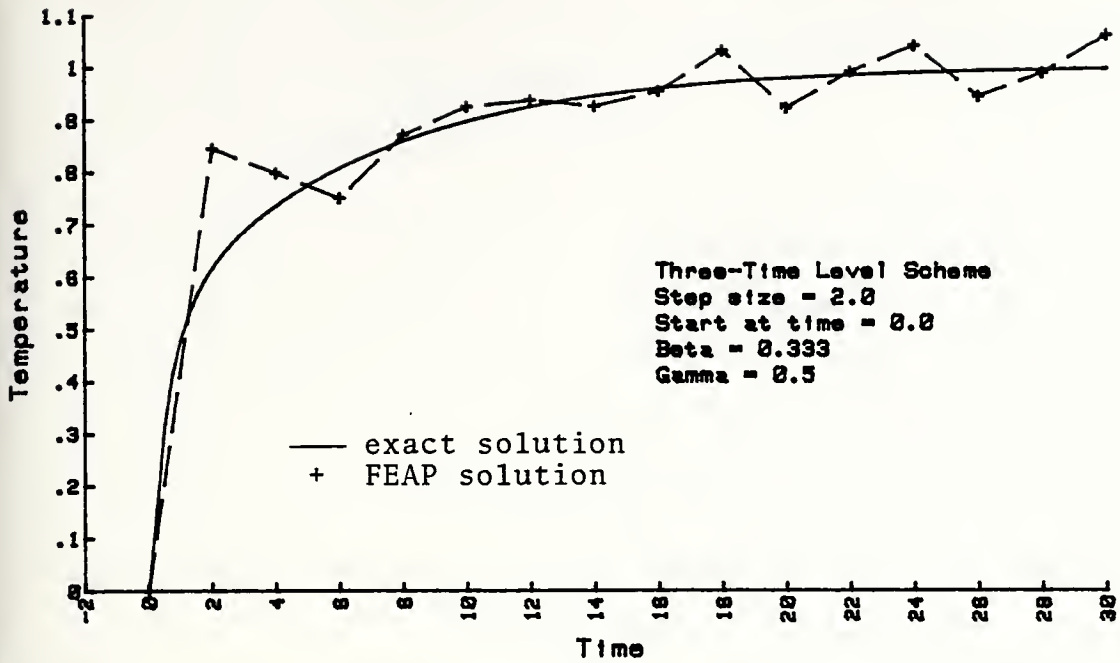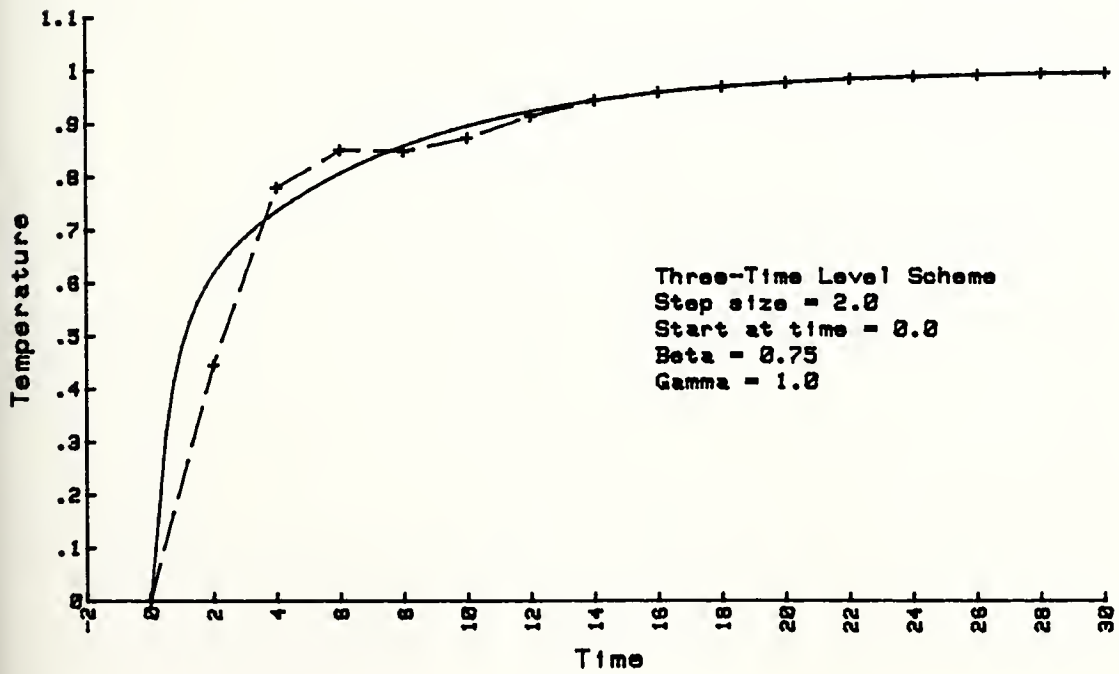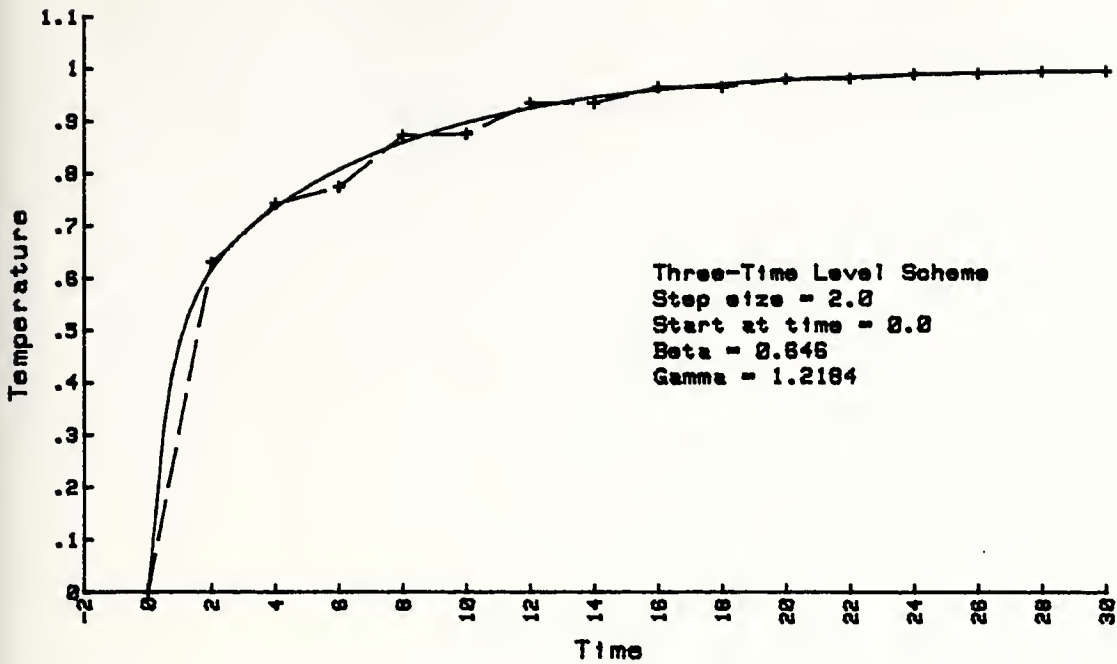
(c)
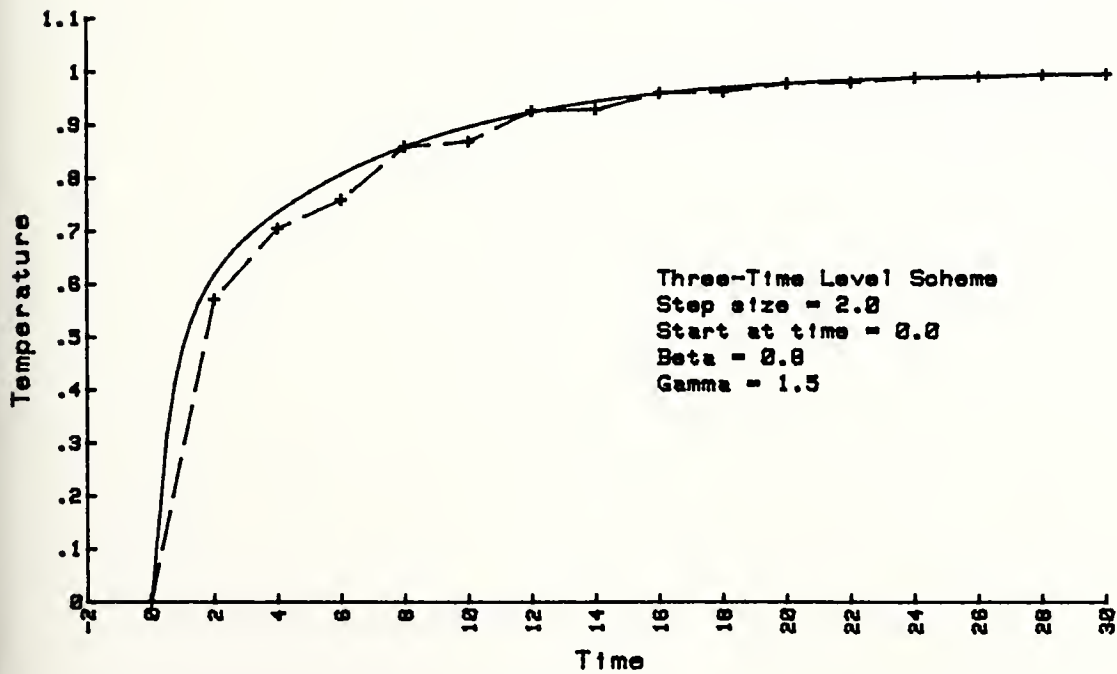
(d)

Figure 13. (continued)

68
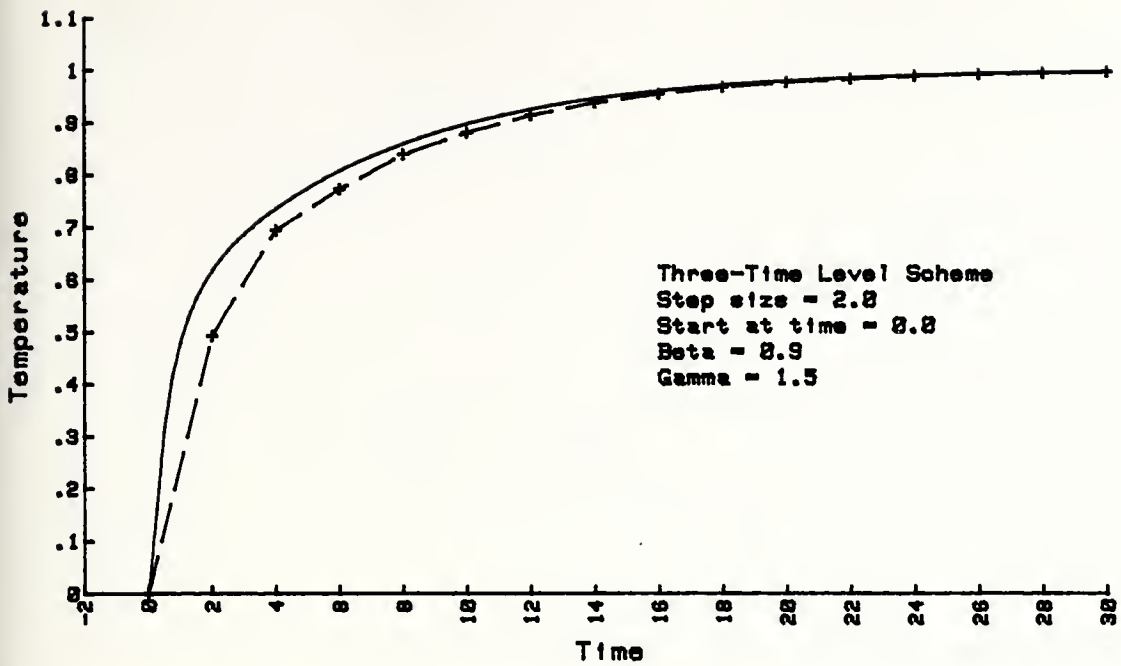
(e)
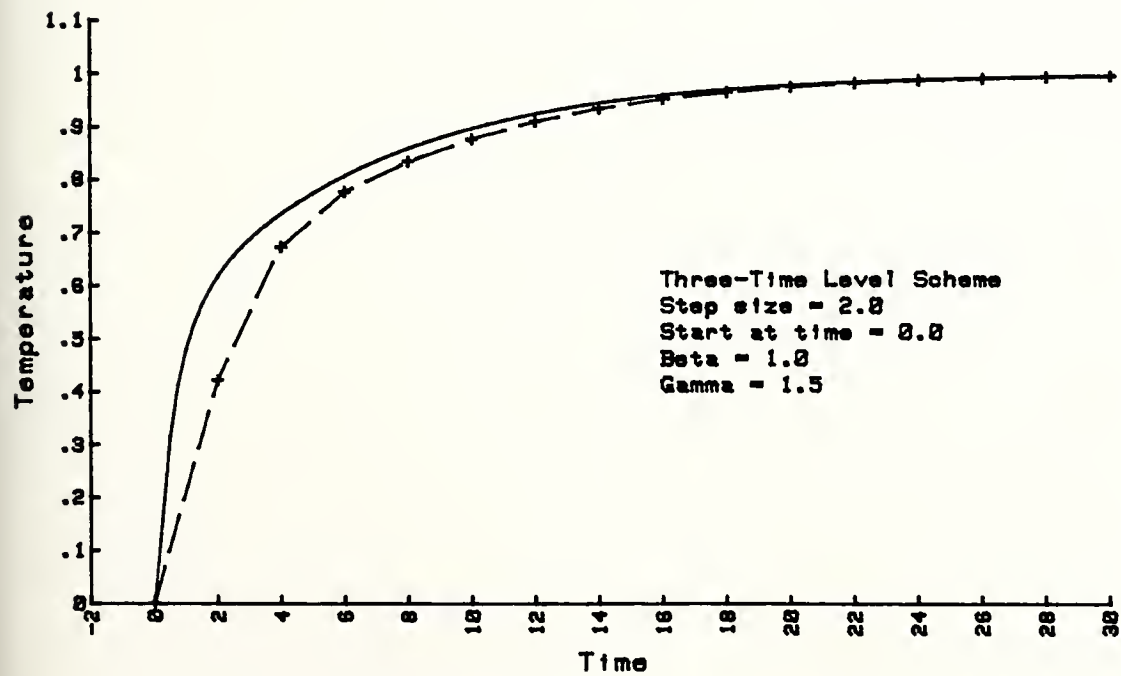
(f)

Figure 13. (continued)

69
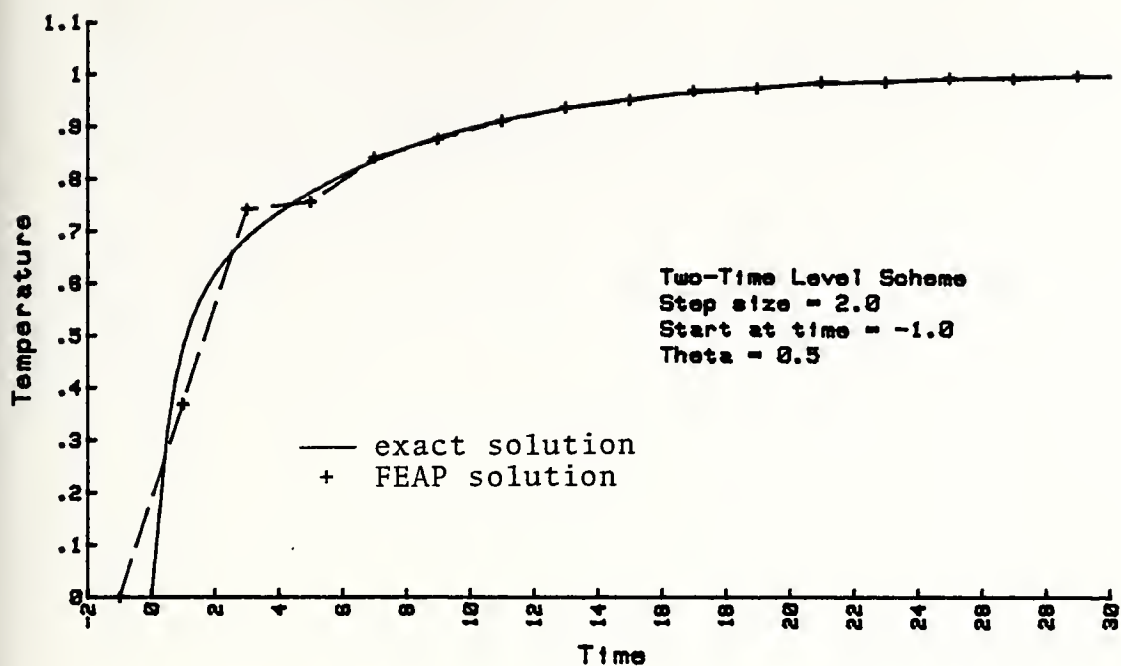
(a)
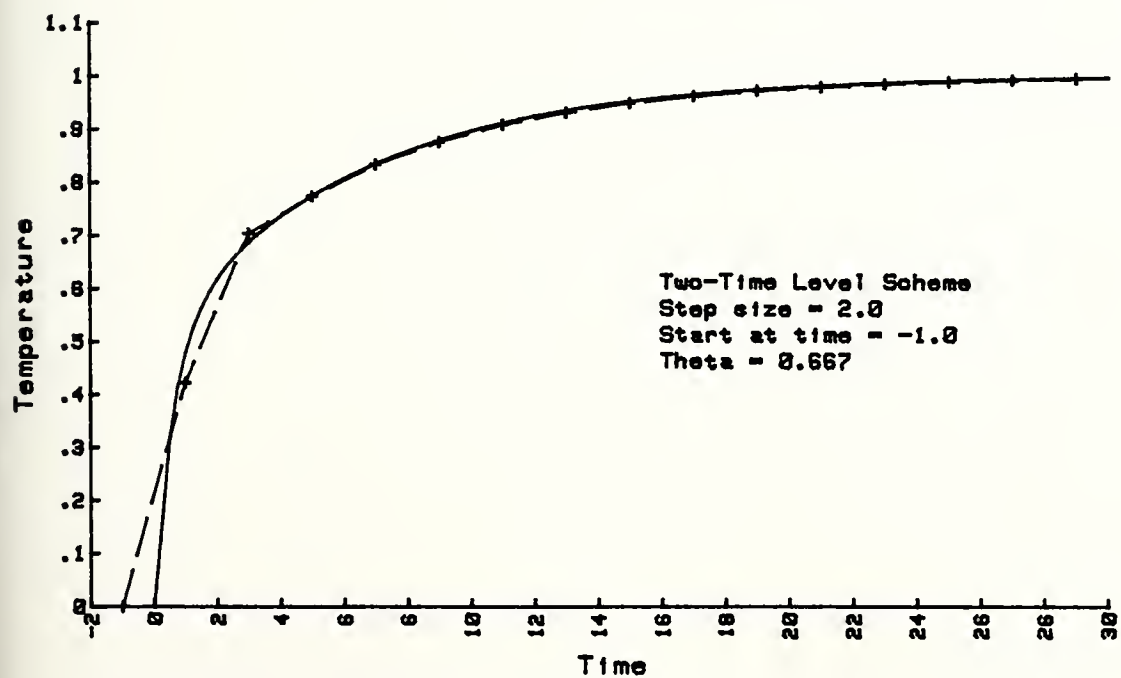
(b)

Figure 14.

(c)

(d)

Figure 14. (continued)

71

(a)

Temperature

1.1
1
.9
.8
.7
.6
.5
.4
.3
.2
.1
0

Three-Time Level Scheme
Step size = 2.0
Start at time = -1.0
Beta = 0.333
Gamma = 0.5

—— exact solution
+ FEAP solution

-2 0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30

Time

(a)

(b)

Temperature

1.1
1
.9
.8
.7
.6
.5
.4
.3
.2
.1
0

Three-Time Level Scheme
Step size = 2.0
Start at time = -1.0
Beta = 0.75
Gamma = 1.0

-2 0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30

Time

(b)

Figure 15.

72
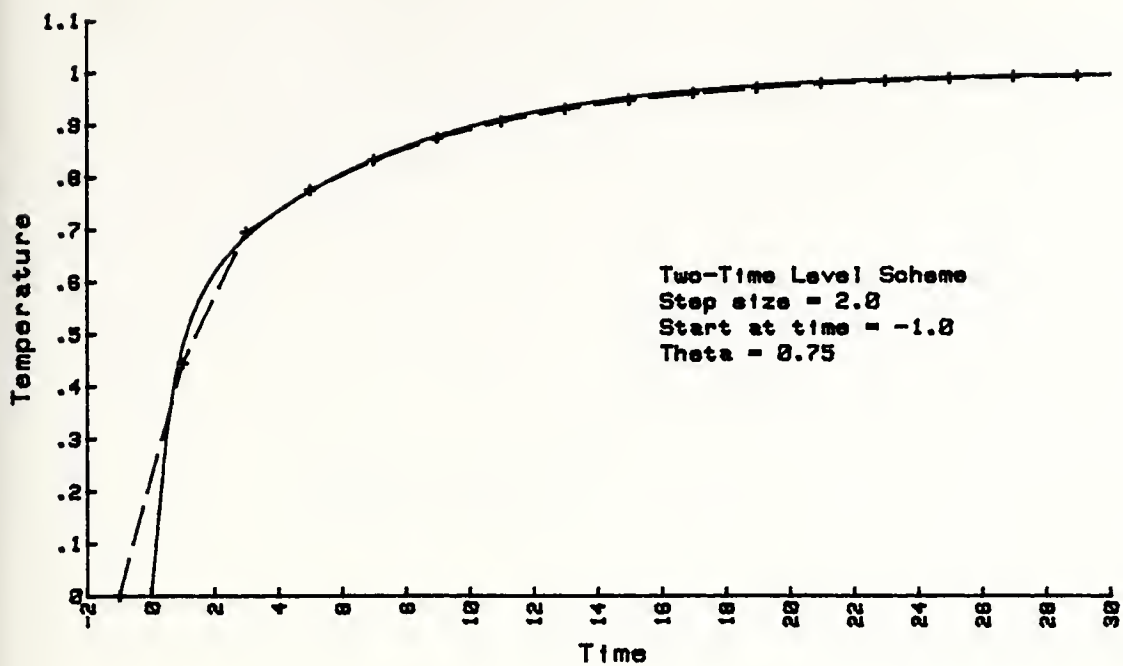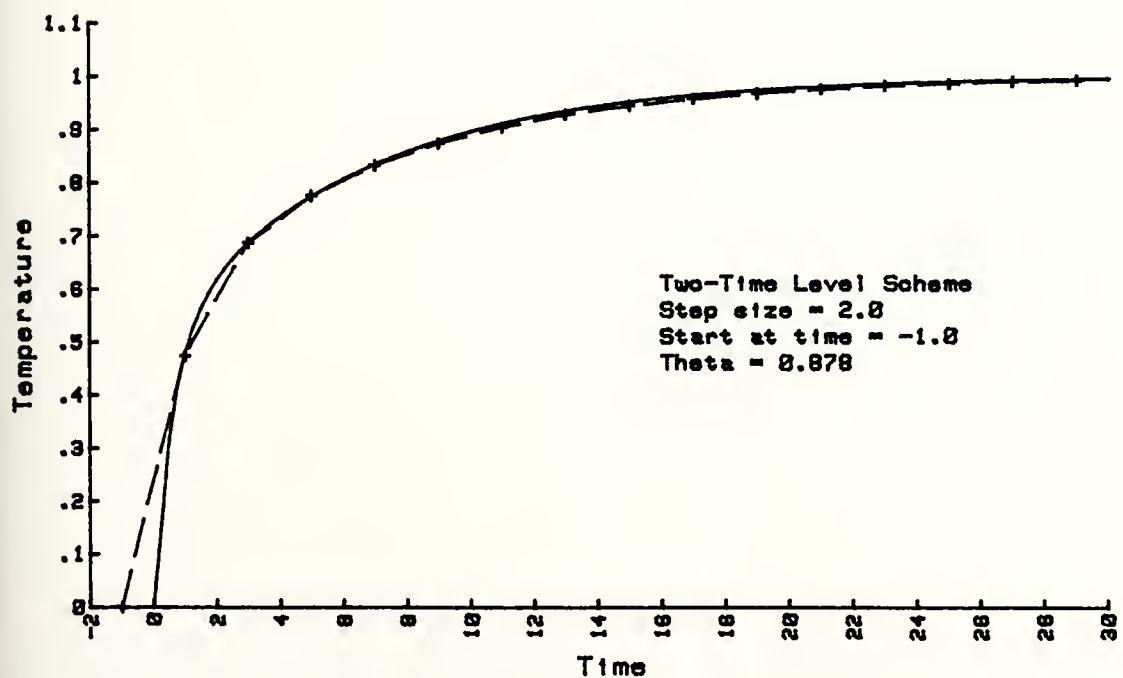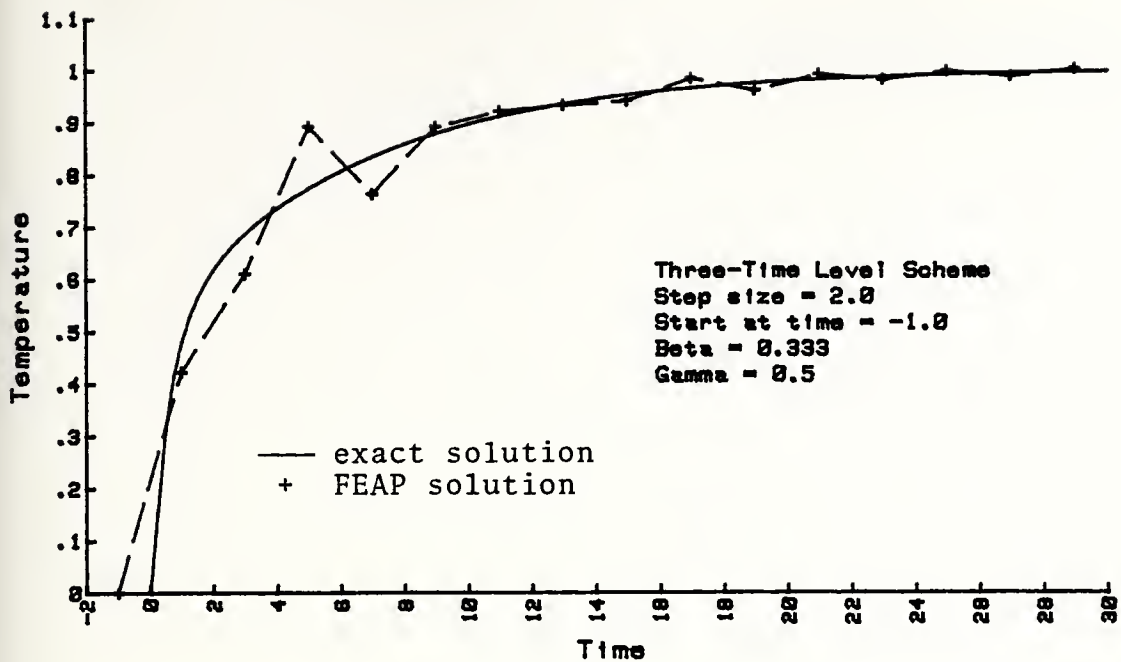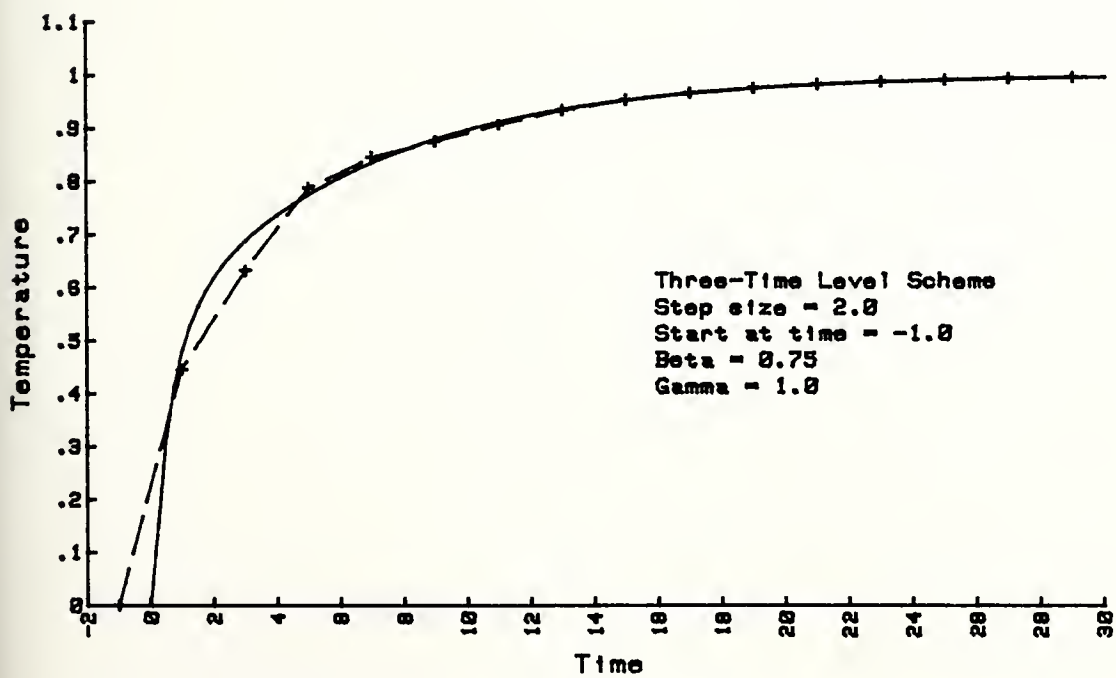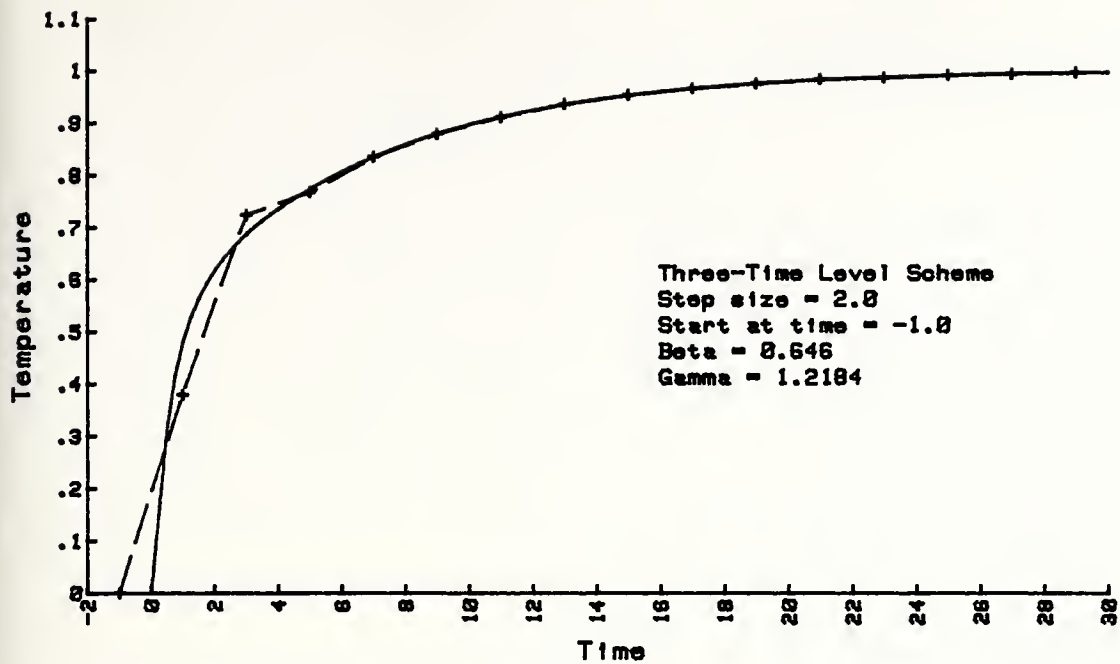
(c)
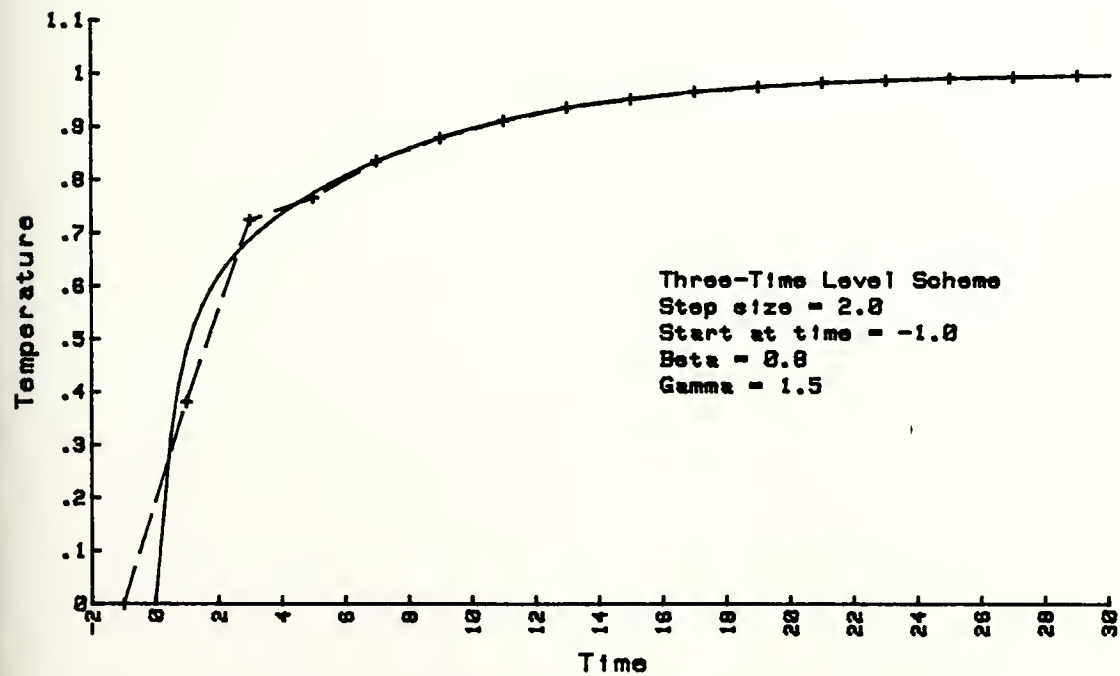
(d)

Figure 15. (continued)

73
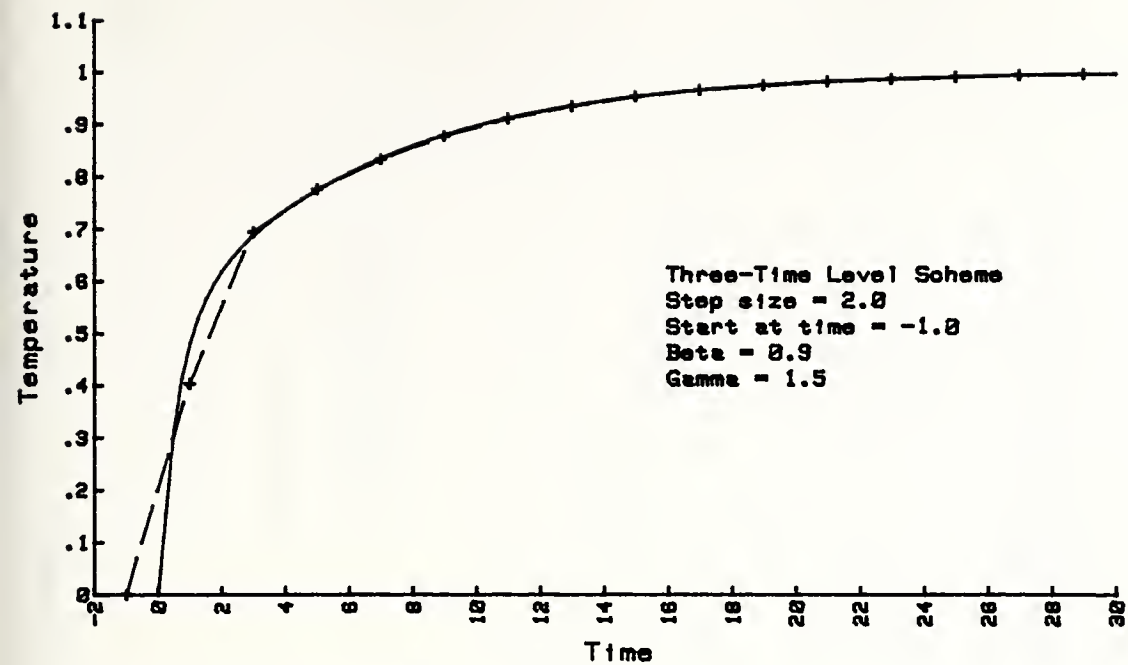
(e)
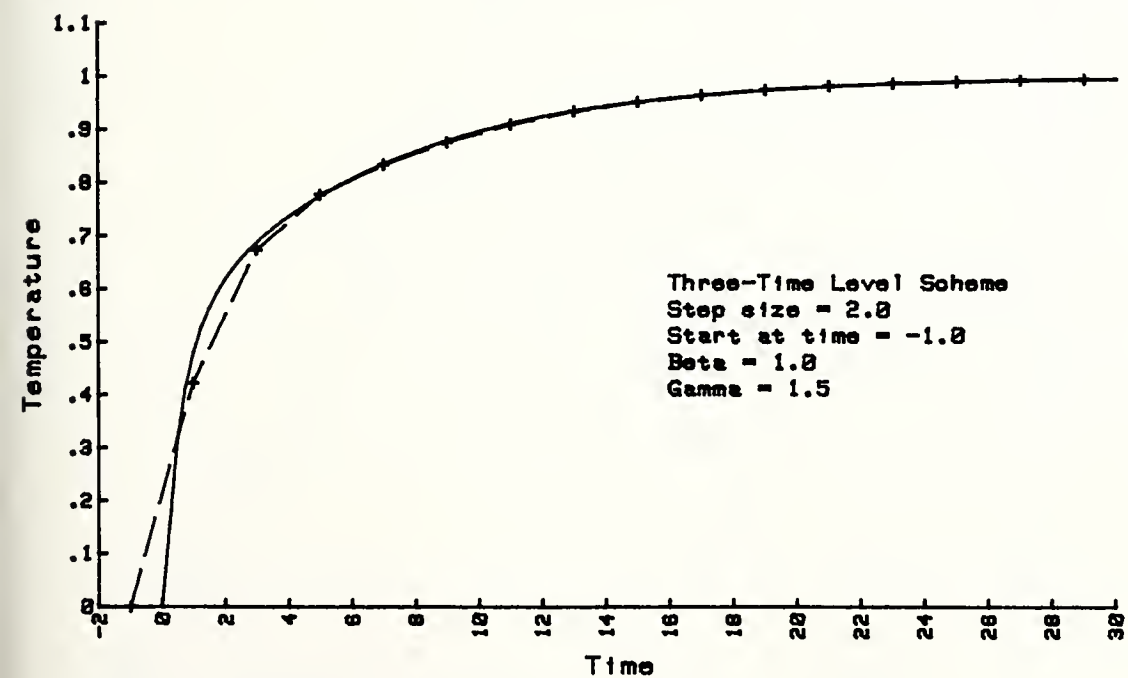
(f)

Figure 15. (continued)

74
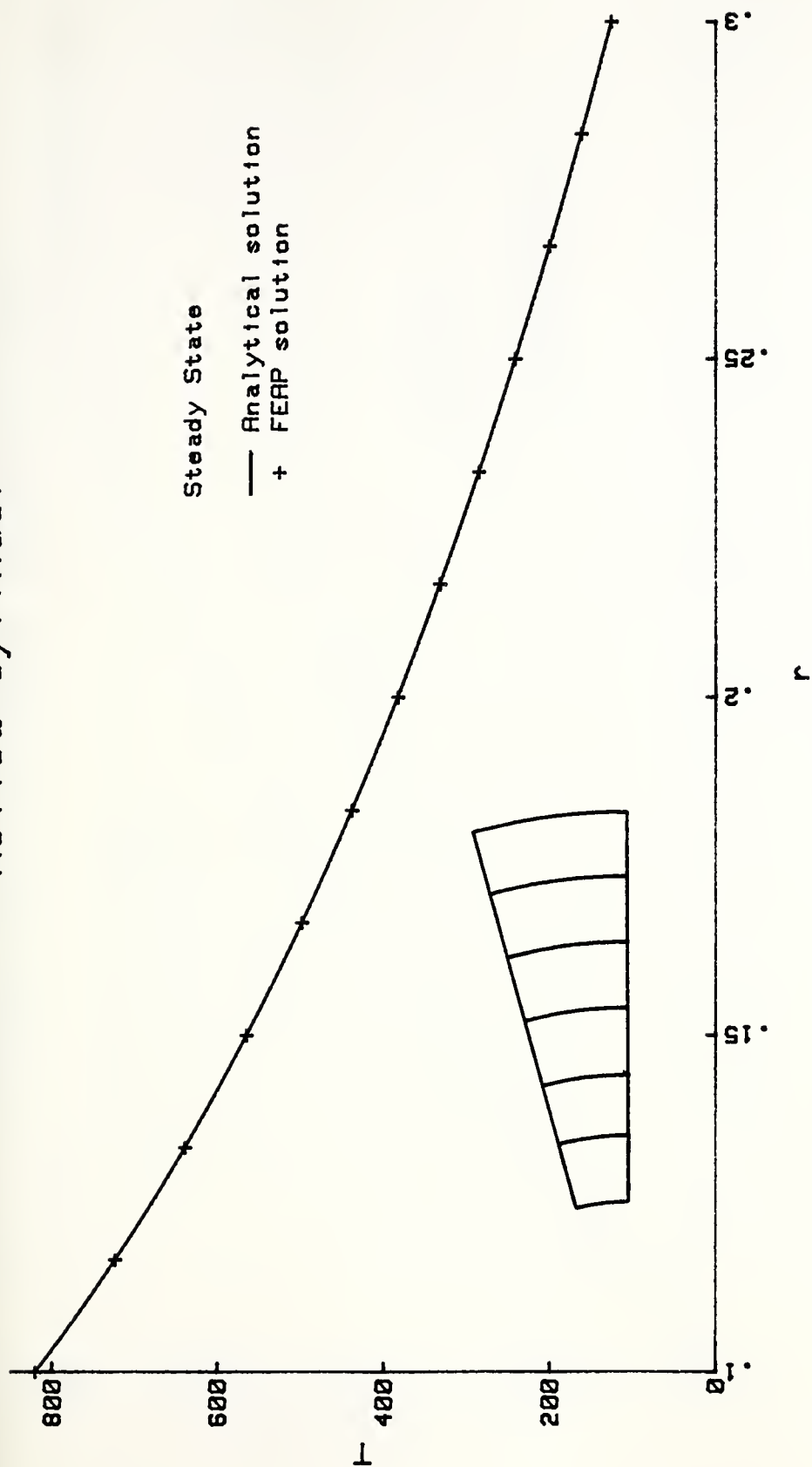
Figure 16. Radial Temperature Distribution in a Hollow Cylinder. The temperature T and the radius r are expressed in degrees Celsius and meters, respectevely.

75

Slab with Variable Conductivity
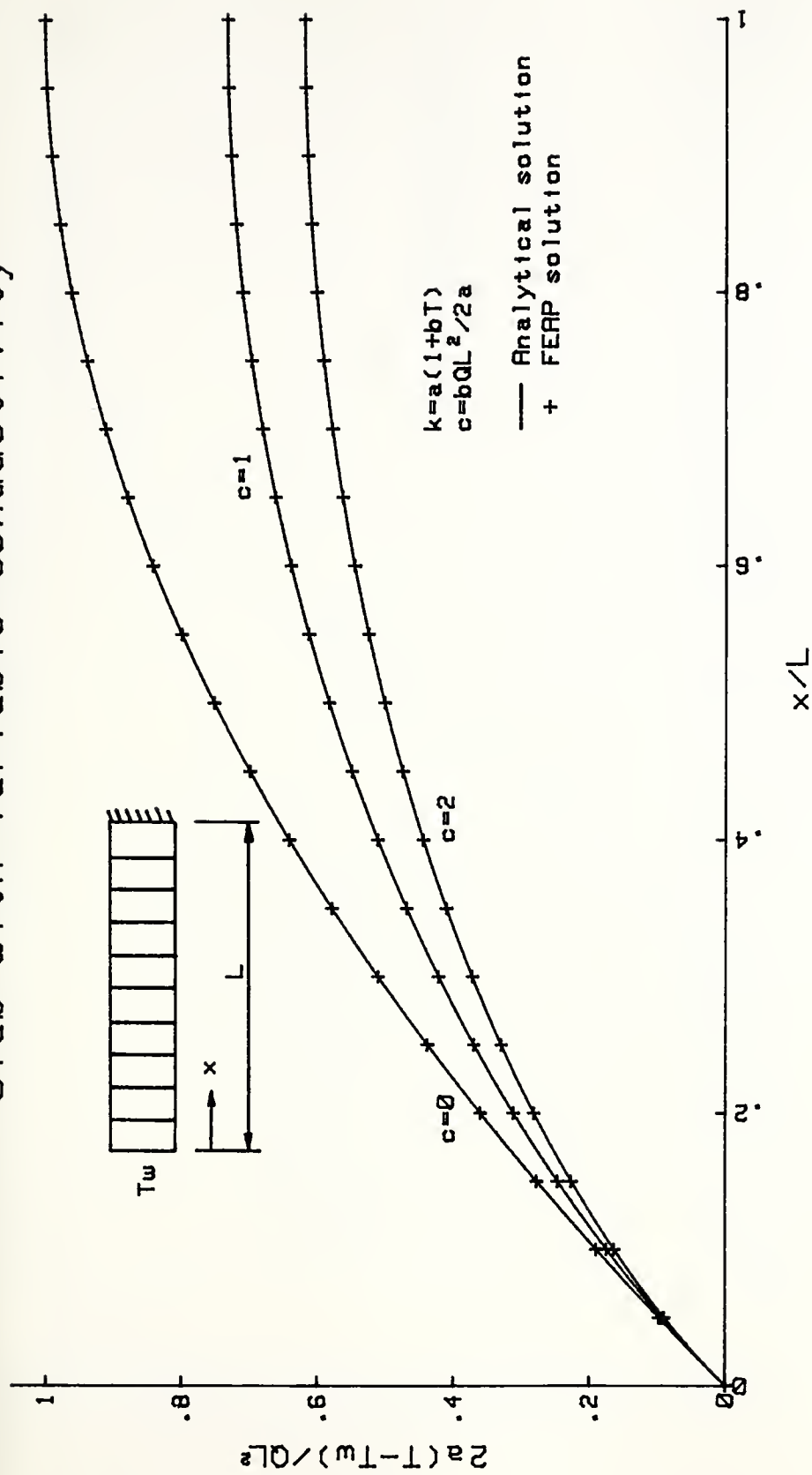
Figure 17. Effect of Linear Conductivity Variation on the Temperature of a Plate with Uniform Heat Generation.

76
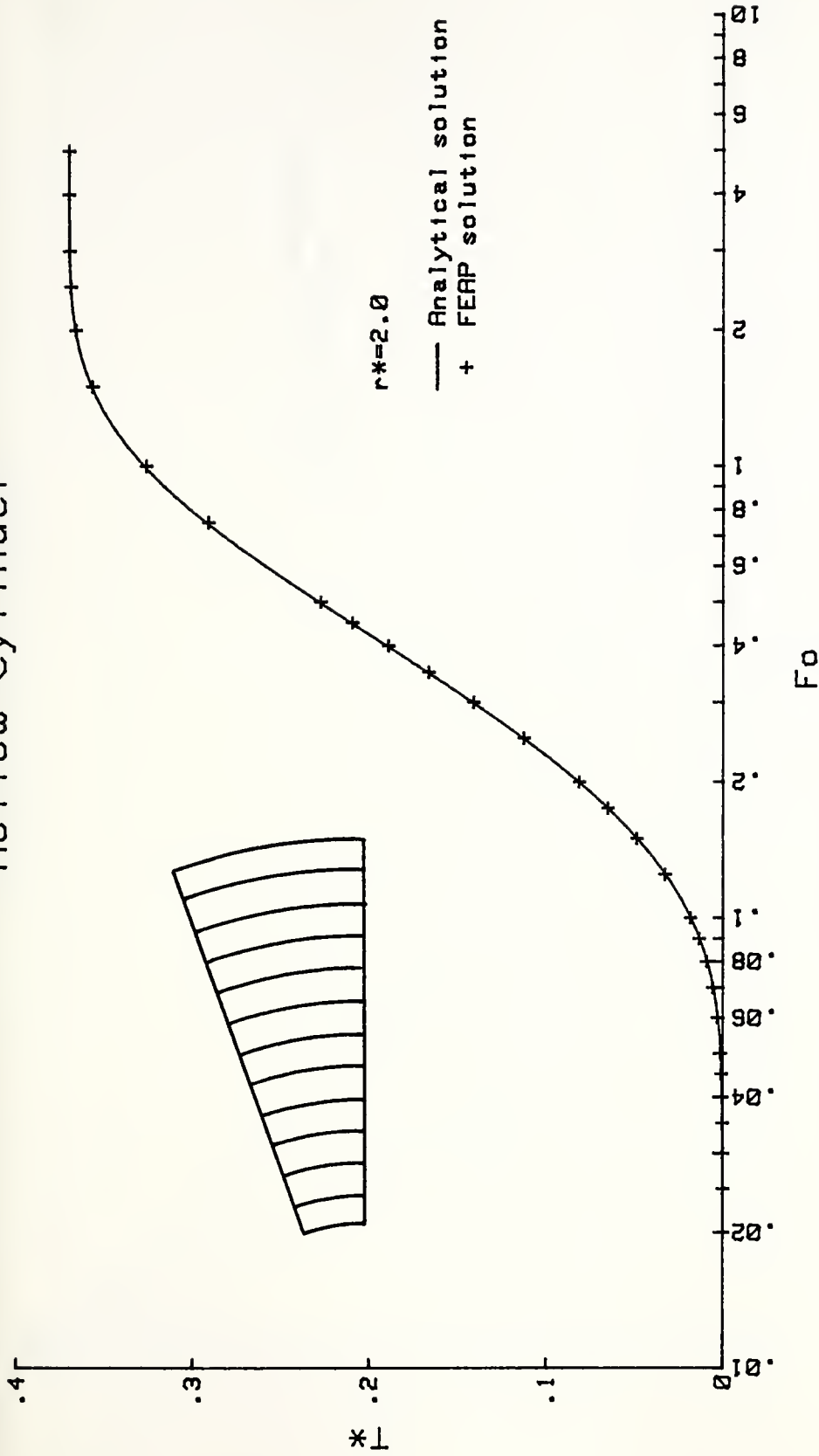
Figure 18. Unsteady Temperature of the Center Interior Points in a
Hollow Cylinder.

77
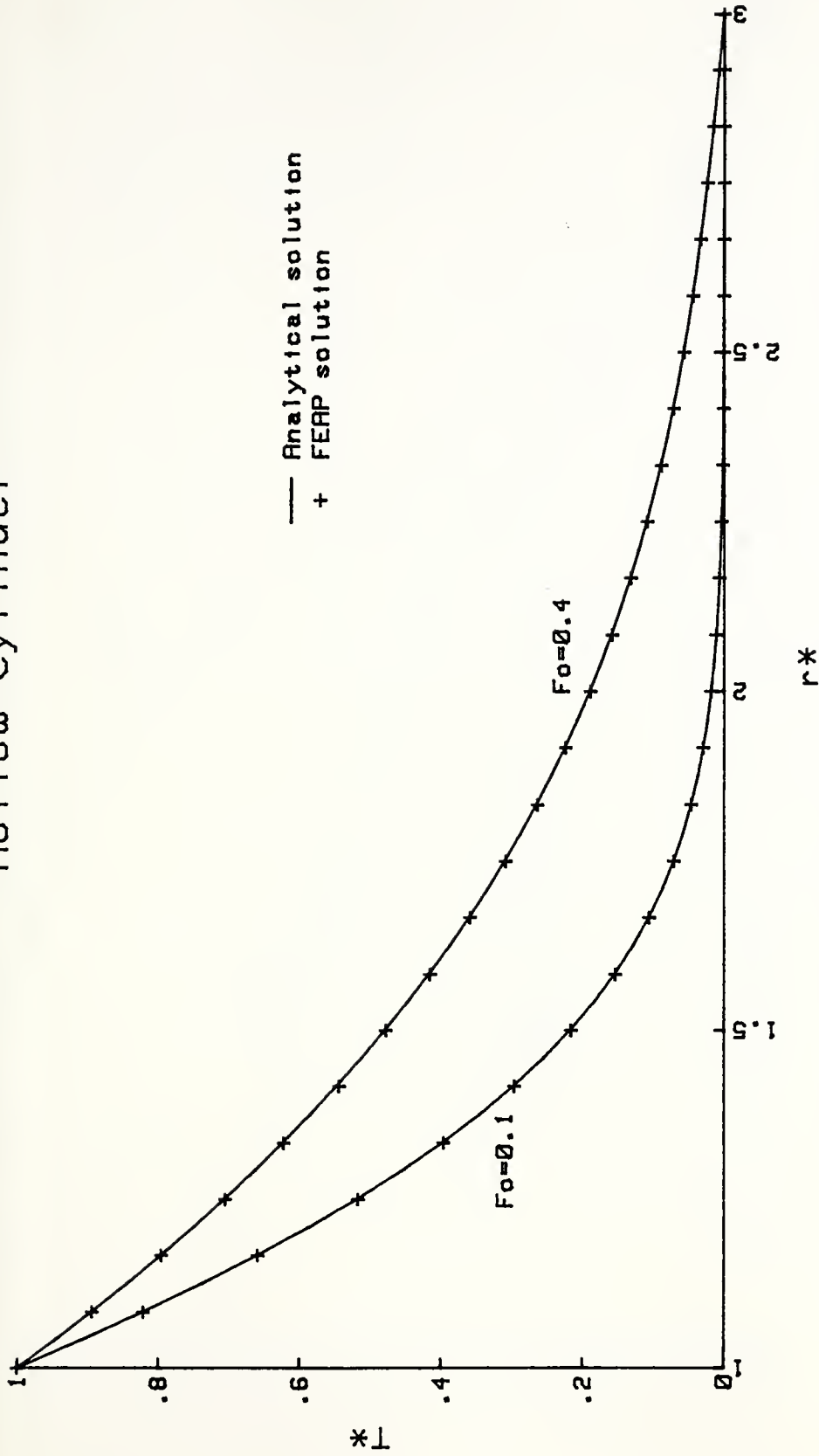
Figure 19. Unsteady Radial Temperature Distribution in a Hollow Cylinder

78
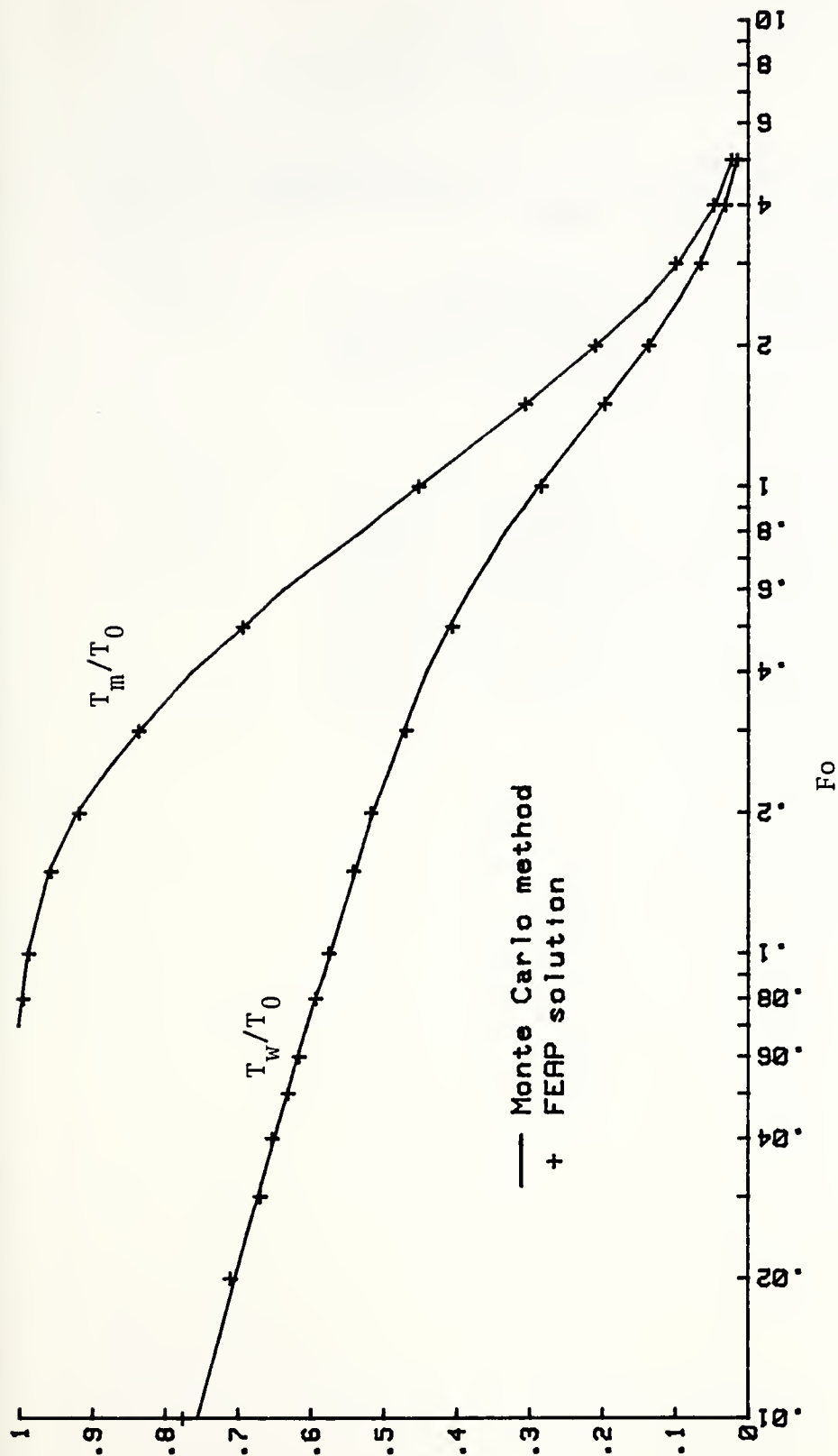
Figure 20. Unsteady Temperature Distribution in a Slab with Radiation/Convection Boundary Conditions.

79

```
C
C                          APPENDIX A
C
C               USER INSTRUCTIONS FOR FEAP
C
C     A.1 TITLE AND CONTROL INFORMATION
C
C
C.... TITLE CARD-FORMAT(20A4)
C     THE TITLE CARD ALSO SERVES AS A START OF PROBLEM CARD.
C     THE FIRST FOUR (4) COLUMNS MUST CONTAIN THE START
C     WORD FEAP
C
C         COLUMNS              DESCRIPTION              VARIABLE
C         1 TO 4        MUST CONTAIN FEAP               TITL(1)
C         5 TO 80       ALPHANUMERIC INFORMA-          TITL(I),I=2,20
C                       TION TO BE PRINTED
C                       WITH OUTPUT AS PAGE
C                       HEADER.
C
C
C.... CONTROL CARD-FORMAT(7I5)
C
C         COLUMNS              DESCRIPTION              VARIABLE
C         1 TO 5        NUMBER OF NODES                 NUMNP
C         6 TO 10       NUMBER OF ELEMENTS              NUMEL
C         11 TO 15      NUMBER OF MATERIAL SETS         NUMMAT
C         16 TO 20      SPATIAL DIMENSION               NDM
C                       (MAXIMUM, UP TO 3)
C         21 TO 25      NUMBER OF UNKNOWNS              NDF
C                       PER NODE (MAXIMUM,
C                       UP TO 6)
C         26 TO 30      NUMBER OF NODES PER             NEN
C                       ELEMENT (MAXIMUM)
C         31 TO 35      NOT USED
```

80

```
C
C        A.2 CATA INFUT: MACRO CCNTROL STATEMENTS
C
C
C.... INFUT MACRC CCNTROL CARCS-FORMAT(A4)
C     THE INFLT CF EACH DATA SEGMENT IS CONTROLLED EY THE
C     VALUE ASSIGNEC TO CC. THE FOLLOWING VALUES ARE ADMI-
C     SSIBLE ANC EACH CC CARC MUST BE IMMEDIATELY FCLLOWED
C     EY THE AFFRCPRIATE DATA.
C
C        CC VALLE          DATA TC BE INPUT
C        COOR              COORDINATE DATA
C        FCLA              CONVERT PCLAR TO CARTESIAN
C                          COCRDINATES
C        SPHE              CONVERT SPHERICAL TO CARTESIAN
C                          COORDINATES
C        ELEM              ELEMENT DATA
C        MATE              MATERIAL DATA
C        BOLN              NODAL BCUNCARY CONDITICN DATA
C        FORC              PRESCRIBED NODAL DISPLACEMENT/FORCE
C                          DATA. FCR HEAT TRANSFER PROBLEMS
C                          THIS IS FRESCRIBED NODAL TEMFERA-
C                          TURE/FLUX CATA
C        TEMP              TEMPERATURE DATA
C        PRIN              PRINT SLBSEQUENT MESH DATA
C                          (DEFAULT MODE)
C        NOPR              DO NCT FRINT SUBSEQUENT MESH CATA
C        PAGE              PRINTED CUTPUT CONTROL
C        ENC               MUST BE LAST CARD IN MESH DATA,
C                          TERMINATES MESH INPUT
C
C     EXCEFT FCR THE ENC CARC THE DATA SEGMENTS CAN BE
C     IN ANY CRDER. IF THE VALLES CF BOUN,FCRC, CR TEMP
C     ARE ZERC, NC INPUT DATA IS REQUIRED.
C
C
C.... COORDINATE CATA - FORMAT(2I5,7F10.0)
C     MUST IMMEDIATELY FOLLCW A COOR MACRO CARD.
C     THE CCCRDINATE CATA CARC CCNTAINS THE NODE NUMBER N
C     AND THE VALLE OF THE CCCRDINATES FOR THE NCDE. CNLY
C     THE VALUES CF (XL(I),I=1,NCM ) ARE USED, WHERE NDM IS
C     THE VALLE INFLT ON THE CCNTROL CARD.
C     NODAL COCRDINATES CAN BE GENERATED ALONG A STRAIGHT
C     LINE DESCRIBED BY THE VALUES ON TWO SUCCESSIVE
C     CARDS. THE VALUE OF THE NCDE NUMBER IS CCMPUTED
C     LSING THE N ANC NG ON THE FIRST CARD TO COMPLLTE
C     THE SECLENCE N, N+NG,N+2*NG, ETC. NG MAY BE INPUT
C     AS A NEGATIVE NUMBER, IF IT HAS INCORRECT SIGN THE
C     SIGN WILL BE CHANGED. NODES NEED NOT BE IN ORDER.
C
C        COLUMNS              DESCRIPTION           VARIABLE
C        1 TC 5           NCDE NUMBER               N
C        6 TC 10          GENERATCR INCREMENT       NG
C        11 TC 2C         X1 COORDINATE             XL(1)=X(1,N)
C        21 TC 3C         X2 COORDINATE             XL(2)=X(2,N)
C        31 TC 40         X3 COORDINATE             XL(3)=X(3,N)
C            TERMINATE WITH BLANK CARD(S)
C
C
C.... FCLAR TC CARTESIAN CCNVERSION DATA-FORMAT(3I5,2F10.0)
C     MUST IMMEDIATELY FOLLCW A POLA MACRO CARD.
C     THE FCLAR CCCRDINATES (R,THETA) MUST BE PREVICUSLY IN-
C     PUT AS CCCRDINATE DATA (MACRC CARD COOR) WHERE R IS X1
C     ANC THETA IS X2. THE NODAL POLAR COORDINATES CF NODES
C     NI, NI+INC, NI+2*INC,..., NE ARE CONVERTEC TC
C     CARTESIAN CCCRDINATES.
C
C        COLUMNS              DESCRIPTION           VARIABLE
C        1 TC 5           FIRST NCDE TC BE          NI
C                         CCNVERTEC
C        6 TC 10          LAST NODE TC BE           NE
```

81

```
C                         CONVERTED
C        11 TO 15         NODAL INCREMENT          INC
C        16 TO 25         X COORDINATE OF THE      XO
C                         ORIGIN
C        26 TO 35         Y COORDINATE OF THE      YO
C                         ORIGIN
C           TERMINATE WITH BLANK CARD(S)
C
C
C
C.... SPHERICAL TO CART. CONVERSION DATA-FORMAT(3I5,3F10.0)
C     MUST IMMEDIATELY FOLLOW A SPHE MACRO CARD.
C     THE SPHERICAL COORDINATES (R,THETA,PHI) MUST BE PRE-
C     VIOUSLY INPUT AS COORDINATE DATA (X1,X2,X3) FOLLOWING
C     MACRO CARD COOR. THETA IS THE ANGLE FORMED BY R AND X
C      AND PHI IS THE ANGLE FORMED BY R AND Z.
C     THE NODAL SPHERICAL COORDINATES OF NODES NI, NI+INC,
C     NI+2*INC,...,NE ARE CONVERTED TO CARTESIAN COORDINATES
C
C        COLUMNS              DESCRIPTION         VARIABLE
C         1 TO 5          FIRST NODE TO BE         NI
C         6 TO 10         LAST NODE TO BE          NE
C                         CONVERTED
C        11 TO 15         NODAL INCREMENT          INC
C        16 TO 2C         X COORDINATE OF THE      XO
C                         ORIGIN
C        26 TO 35         Y COORDINATE OF THE      YO
C                         ORIGIN
C        36 TO 45         Z COORDINATE OF THE      ZO
C                         ORIGIN
C           TERMINATE WITH BLANK CARD(S).
C
C
C
C.... ELEMENT DATA - FORMAT(16I5)
C     MUST IMMEDIATELY FOLLOW AN ELEM MACRO CARD.
C     THE ELEMENT DATA CARD CONTAINS THE ELEMENT NUMBER,
C     MATERIAL SET NUMBER (WHICH ALSO SELECTS THE ELEMENT
C     TYPE,SEE MATERIAL PROPERTY DATA), AND THE SEQUENCE
C     OF NODES CONNECTED TO THE ELEMENT.IF THERE ARE LESS
C     THAN NEN NODES EITHER LEAVE THE APPROPRIATE FIELDS
C     BLANK OR PUNCH ZEROS.
C     ELEMENTS MUST BE IN ORDER. IF ELEMENT CARDS ARE
C     OMITTED THE ELEMENT DATA WILL BE GENERATED FROM
C     THE PREVIOUS ELEMENT WITH THE SAME MATERIAL NUMBER
C     AND THE NODES ALL INCREMENTED BY LX ON THE PREVIOUS
C     ELEMENT.GENERATION TO THE MAXIMUM ELEMENT NUMBER
C     OCCURS WHEN A BLANK CARD IS ENCOUNTERED.
C
C        COLUMNS              DESCRIPTION         VARIABLE
C         1 TO 5          ELEMENT NUMBER           L
C         6 TO 10         MATERIAL SET NUMBER      IX(NEN1,L)
C        11 TO 15         NODE 1 NUMBER            IX(1,L)
C        16 TO 20         NODE 2 NUMBER            IX(2,L)
C        ETC.                 .
C        ETC.            NODE NEN NUMBER           IX(NEN,L)
C        ETC.            GENERATION INCREMENT      LX
C           TERMINATE WITH BLANK CARD(S).
C
C
C
C.... MATERIAL PROPERTY DATA - FORMAT(I5,4X,I1,17A4)
C     MUST IMMEDIATELY FOLLOW A MATE MACRO CARD.
C
C        COLUMNS              DESCRIPTION         VARIABLE
C         1 TO 5          MATERIAL SET NUMBER      MA
C         6 TO 9
C            10           ELEMENT TYPE NUMBER      IEL
C        11 TO 78         ALPHANUMERIC INFORMATION XHED
C                         TO BE OUTPUT
C     EACH MATERIAL CARD MUST BE FOLLOWED IMMEDIATELY
```

82

```
C                BY THE MATERIAL PROPERTY DATA REQUIRED FOR THE
C                ELEMENT TYPE IEL BEING USED,E.G.,SEE SECTION A.4
C                AND A.5 FOR HEAT TRANSFER ELEMENTS.
C
C
C.... BOUNDARY RESTRAINT DATA - FORMAT(16I5)
C     MUST IMMEDIATELY FOLLOW A BOUN MACRO CARD.
C     FOR EACH NODE WHICH HAS AT LEAST ONE DEGREE OF FREEDOM
C     WITH A SPECIFIED DISPLACEMENT,A BOUNDARY CONDITION
C     CARD MUST BE INPUT. THE CONVENTION USED FOR BOUNDARY
C     RESTRAINTS IS
C          .EQ.0  NO RESTRAINT,FORCE SPECIFIED
C          .NE.0  RESTRAINED,DISPLACEMENT SPECIFIED
C     VALUES OF FORCE OR DISPLACEMENT INPUT IN FORC
C
C        COLUMNS              DESCRIPTION              VARIABLE
C        1 TO 5          NODE NUMBER                   N
C        6 TO 10         GENERATION INCREMENT          NX
C        11 TO 15        DOF 1 BOUNDARY CODE           IDL(1)=ID(1,N)
C        16 TO 20        DOF 2 BOUNDARY CODE           IDL(2)=ID(2,N)
C          ETC.                    .                       .
C          ETC.          DOF NDF BOUNDARY CODE         IDL(NDF)=
C                                                         ID(NDF,N)
C        WHEN GENERATING BOUNDARY CONDITION CODES FOR
C        SUBSEQUENT NODES IDL IS SET TO ZERO IF IT WAS
C        INPUT .GE.0, AND IS SET TO -1 IF INPUT NEGA-
C        TIVE. ALL DEGREES OF FREEDOM WITH NON-ZERO
C        CODES ARE ASSUMED FIXED.
C             TERMINATE WITH BLANK CARD(S).
C
C
C.... NODAL FORCED BOUNDARY VALUE DATA - FORMAT(2I5,7F10.0)
C     MUST IMMEDIATELY FOLLOW A FORC MACRO CARD.
C     FOR EACH NODE WHICH HAS A NON-ZERO NODAL FORCE OR
C     DISPLACEMENT A FORCE CARD MUST BE INPUT OR GENERATED.
C     GENERATION IS THE SAME AS FOR CO-ORDINATE DATA. THE
C     VALUE SPECIFIED IS A FORCE IF THE CORRESPODING RES-
C     TRAINT CODE IS ZERO AND A DISPLACEMENT IF THE CORRES-
C     PONDING RESTRAINT CODE IS NON-ZERO.
C
C        COLUMNS              DESCRIPTION              VARIABLE
C        1 TO 5          NODE NUMBER                   N
C        6 TO 10         GENERATION INCREMENT          NG
C        11 TO 20        DOF 1 FORCE (DISP.)           XL(1)=F(1,N)
C        21 TO 30        DOF 2 FORCE (DISP.)           XL(2)=F(2,N)
C          ETC.                    .                       .
C          ETC.          DOF NDF FORCE (DISP.)         XL(NDF)=
C                                                         F(NDF,N)
C           TERMINATE WITH A BLANK CARD.
C
C
C.... TEMPERATURE DATA CARD - FORMAT(2I5,F10.0)
C     MUST IMMEDIATELY FOLLOW A TEMP MACRO CARD.
C     NOT USED IN HEAT TRANSFER PROBLEMS.
C     FOR EACH NODE WHICH HAS A NON-ZERO TEMPERATURE THE
C     VALUE MUST BE INPUT. GENERATION OF VALUES CAN BE
C     PERFORMED AS DESCRIBED FOR CO-ORDINATES.
C
C        COLUMNS              DESCRIPTION              VARIABLE
C        1 TO 5          NODE NUMBER                   N
C        6 TO 10         GENERATION INCREMENT          NG
C        11 TO 20        NODAL TEMPERATURE             XL(1)=T(N)
C           TERMINATE WITH BLANK CARD(S).
C
C
C.... PAGE CONTROL DATA - FORMAT(A1)
C     MUST IMMEDIATELY FOLLOW A PAGE MACRO CARD
C     THE VALUE IN COLUMN 1 CONTROLS THE PRINTED OUTPUT
```

```
C         ACCORDING TC THE FOLLCWING CONVENTION:
C              1 TITLE IS WRITEN AT THE TOP OF A NEW PAGE
C              0 THE PRINTED OUTFLT IS CONTINUOUS WITHOUT
C                SKIFFING PAGES (CEFALLT MODE).
```

```
C        A.3 PROBLEM SOLUTION: MACRO PROGRAMMING COMMANDS
C
C
C.... MACRO PROGRAMMING COMMANDS - FORMAT(2(A4,1X),F10.0)
C     FOLLOWING IS A LIST OF MACRO INSTRUCTIONS WHICH
C     CAN BE USED TO CONSTRUCT SOLUTION ALGORITHMS. THE
C     FIRST INSTRUCTION MUST BE A CARD WITH MACR IN
C     COLUMNS 1 TO 4. THE INDICATED ISW VALUE IS USED BY
C     EACH ELEMENT ROUTINE TO PERFORM THE APPROPRIATE
C     COMPUTATIONS.
C
C     COLUMNS  COLUMNS  COLUMNS              DESCRIPTION
C     1 TO 4   6 TO 9   11 TO 20
C     CHEC                          PERFORM CHECK OF MESH (ISW=2)
C     CMAS                          CONSISTENT MASS FORMULATION
C                                     (ISW=5)
C     CONV                          DISPLACEMENT CONVERGENCE TEST
C     DATA                          READ DATA **MACRO COMMANDS. A
C                                   **MACRO CARD IS INSERTED AS DATA
C                                   FOLLOWING THE MACRO PROGRAM
C     DISP              N           PRINT NODAL DISPLACEMENTS EVERY
C                                   N STEPS IN LOOP
C  **DT                V           SET TIME INCREMENT TO VALUE V
C     EIGE                          COMPUTE DOMINANT EIGENVALUE AND
C                                   VECTOR OF CURRENT MASS AND
C                                   SYMMETRIC STIFFNESS
C     EXCD                          EXPLICIT CENTERED DIFFERENCE
C                                   INTEGRATION OF EQUATIONS OF
C                                   MOTION USING LUMPED MASS. FIRST
C                                   CALL RESERVES MEMORY ONLY.
C     FORM                          FORM RIGHT HAND SIDE OF EQUA-
C                                   TIONS    (ISW=6)
C     LMAS                          LUMPED MASS FORMULATION (ISW=5)
C     LOOP              N           LOOP N TIMES ALL INSTRUCTIONS
C                                   BETWEEN  MATCHING  NEXT
C                                   INSTRUCTION.
C     MESH                          INPUT MESH CHANGES (MUST NOT
C                                   CHANGE BOUNDARY RESTRAINTS).DATA
C                                   FOLLOWS MACRO PROGRAM. SEE
C                                   SECTION A.2.
C     NEXT                          END OF LOOP INSTRUCTION
C     ODE1     INIT                 INPUT INITIAL CONDITIONS AND
C                                   CONSTANTS OF INTEGRATION FOR
C                                   FIRST ORDER ORDINARY DIFFEREN-
C                                   TIAL EQUATION SOLUTION (DATA
C                                   FOLLOWS MACRO PROGRAM)
C     ODE1     LINE                 TWO-POINT INTEGRATION SCHEME
C     ODE1     QUAD                 THREE-POINT INTEGRATION SCHEME
C     PROP              1           INPUT PROPORTIONAL LOAD TABLE
C                                   (DATA FOLLOWS MACRO PROGRAM)
C     REAC                          COMPUTE NODAL REACTIONS(ISW=6)
C     SOLV                          SOLVE TANGENT EQUATIONS.UPDATE
C                                   NODAL DISPLACEMENTS.
C     STRE              N           PRINT ELEMENT VARIABLES (E.G.,
C                                   STRESSES) EVERY N STEPS IN LOOP
C                                     (ISW=4)
C     TANG                          SYMMETRIC TANGENT STIFFNESS
C                                   FORMULATION (ISW=3)
C     TIME                          ADVANCE TIME BY DT VALUE
C     *TIM                          INDICATE EXECUTION TIME FOR
C                                   EVERY MACRO INSTRUCTION
C  **TOL                V           SET SOLUTION CONVERGENCE TOLE-
C                                   RANCE TO VALUE V
C     UTAN                          UNSYMMETRIC TANGENT STIFFNESS
C                                   FORMULATION (ISW=3)
C     END                           END OF MACRO PROGRAM COMMANDS.
C                                   DATA FOR PROGRAM FOLLOWS IN
C                                   ORDER OF USE
C
C
C.... FIRST ORDER O.D.E. SOLVER DATA
```

```
C      MUST FCLLCW THE MACRC CARD END AND CORRESPONDS TO
C      THE MACRC INSTRUCTICN CCE1 INIT. THE FIRST CARD
C      INPUTS TFE CCNTROL INFCRMATION AND THE NODAL INITIAL
C      CISPLACEMENTS ARE INPLT CN THE FOLLOWING CARDS.
C      NC ALTCMATIC TIME STEF ACJLSTMENT IS PERFCRMEC IF
C      CUMAX CR CLMIN ARE LEFT BLANK OR ZERO.
C.... CONTRCL CARC - FORMAT(4F10.0)
C
C         COLUMNS              DESCRIPTION              VARIABLE
C          1 TC 1C        INTEGRATION PARAMETER         C5
C                         THETA FCR TWC PCINT
C                         SCHEME (DEFALLT VALUE
C                         THETA=2/3)
C         11 TC 2C        INTEGRATICN FARAMETER         C1
C                         GAMMA FCR THREE POINT
C                         SCHEME
C                         (DEFAULT GAMMA=1.5)
C         21 TC 3C        INTEGRATICN PARAMETER         C2
C                         EETA FCR TFREE PCINT
C                         SCHEME
C                         (CEFAULT BETA=.8)
C         31 TC 40        MAXIMUM CISPLACEMENT          DUMAX
C                         ALLOWED
C         41 TC 5C        MINIMUM CISPLACEMENT          DUMIN
C                         ALLOWED
C.... INITIAL CCNCITION DATA - FCRMAT(2I5,7F10.0)
C      FOR EACH NCDE WHICH HAS A NON-ZERO INITIAL DISFLACE-
C      MENT A CARC MUST BE INPUT CR GENERATED. THE GENERATION
C      IS THE SAME AS FOR CCCRCINATE CATA.
C
C         COLUMNS              DESCRIPTICN              VARIABLE
C          1 TC 5         NCDE NUMBER                   N
C          6 TO 1C        GENERATICN INCREMENT          NG
C         11 TC 10        CCF 1 CISPLACEMENT            XL(1)=U(1+N-1)
C          ETC.              •
C          ETC.          COF NDF CISPLACEMENT           XL(NCF)=
C
C          TERMINATE WITH ELANK CARD(S).
C
C
C.... FRCFCRTICNAL LCAC CARC - FCRMAT(2I5,6F10.0)
C      A SIMPLIFIEC PROPORTICNAL LOADING IS PERMITTEC WITH
C         PROF = A1 + A2*TIME + A3*(SIN(A4*TIME + A5))**L
C      WHERE THE CCEFFICIENTS ARE INPUT ON A CATA CARC
C      FOLLOWING TFE END MACRC CARC ACCORDING TO THE
C      FCLLCWINC TAELE.
C
C         COLUMN               CESCRIFTION
C          6 TC 10        L
C         11 TC 2C        MINIMLM TIME FOR WHICH PRCP IS
C                         COMPUTEC
C         21 TC 3C        MAXIMLM TIME FCR WHICH FRCF IS
C                         COMPUTED
C         31 TC 40        A1
C         41 TC 5C        A2
C         51 TC 6C        A3
C         61 TC 7C        A4
C         71 TC 8C        A5
C
```

```
C      A.4 TWC CIMENSIONAL HEAT TRANSFER ELEMENT DATA
C
C
C                                    +T
C                                    +
C                                    +
C                                    +
C                       4----------7----------3
C                       |           +          |
C                       |           +          |
C                       |           +          |
C                       |           +          |      S
C                       8           S++++++++++6+++
C                       |                      |
C                       |                      |
C                       |                      |
C                       |                      |
C                       1----------5----------2
C
C      MLST FCLLCW THE MATE MACRC CARD.
C      THE TWC CIMENSIONAL HEAT TRANSFER ELEMENT IS CALLED
C      ELMTO2 AND THUS THE ELEMENT TYPE NUMBER IN CCLLMN 10
C      CF EACH MATERIAL SET NUMBER CARD MUST BE 2 WHEN THIS
C      ELEMENT IS REQUESTED. THE SECOND CARD GIVES GENERAL
C      INFORMATICN AND IS PREPARED AS FOLLOWS:
C
C.... GENERAL INFCRMATION CATA - FORMAT(5F10.0,4I5)
C
C      COLUMNS            DESCRIPTION               VARIABLE
C       1 TC 1C      CONDUCTIVITY IN X DIR.          D(1)
C                    (IGNORED IF TEMPERATURE
C                    CEPENDENT)
C      11 TC 2C      CONDUCTIVITY IN Y DIR.          C(2)
C                    (IGNOREC IF TEMPERATURE
C                    CEPENDENT)
C      21 TC 30      SPECIFIC HEAT (IGNCRED          D(3)
C                    IF TEMPERATURE DEPEN-
C                    CENT)
C      31 TC 4C      DENSITY (IGNCRED IF             D(4)
C                    TEMPERATURE CEPENDENT)
C      41 TC 5C      HEAT GENERATICN PER             C(5)
C                    UNIT VOLUME (IGNORED
C                    IF TEMPERATURE CEPEN-
C                    DENT)
C      51 TC 55      NUMBER CF INTEGRATION           NGP
C                    FCINTS FER CIRECTION
C                    (DEFALLT 4)
C      56 TC 6C      GEOMETRY TYPE                   KAT
C                    .EQ.2 FCR AXISYMMETRY
C                    .NE.2 FCR PLANE GEOMETRY
C      61 TC 65      TOTAL NUMBER OF LINES           NLBC
C                    WITH SPECIFIED BCUN-
C                    CARY CONCITICNS IN
C                    ELEMENTS WITH THE SAME
C                    MATERIAL SET NUMBER
C      66 TC 7C      CODE TO INCICATE IF ANY         INOL
C                    CF THE MATERIAL PRCPER-
C                    TIES IS TEMPERATURE DE-
C                    CEPENDENT.
C                    .EQ.0 IF ALL MATERIAL
C                     PROPERTIES ARE CONSTANT
C                    .NE.0 IF ANY OF THE MATE-
C                     RIAL PRCPERTIES IS TEM-
C                     PERATURE CEPENDENT
C
C      THE FCLLCWING CATA ARE RECUIRED IF ANY OF THE MATERIAL
C      PRCPERTIES IS TEMPERATURE CEPENDENT. THEIR CRCER IS
C      CRUCIAL AND THCSE CARCS WHICH CORRESPOND TO CCNSTANT
C      MATERIAL FRCPERTIES MLST BE OMITTED.
```

87

```
C....  MATERIAL PROPERTY CODE -FORMAT(4I5)
C      THIS CARD IS ALLWAYS REQUIRED IF INOL.NE.0 AND MUST
C      IMMEDIATELY FOLLOW THE GENERAL INFORMATION DATA CARD.
C
C         COLUMNS            DESCRIPTION         VARIABLE
C         1 TO 5    .EQ.0 CONSTANT CONDUCTI-  IKX
C                   VITY IN X DIR.
C                   .NE.0 TEMP. DEP. KX
C         6 TO 10   .EQ.0 CONSTANT CONDUCTI-  IKY
C                   VITY IN Y DIR.
C                   .NE.0 TEMP. DEP. KY
C         11 TO 15  .EQ.0 CONSTANT HEAT CA    IRCC
C                   PACITY (SPECIFIC HEAT *
C                   DENSITY)
C                   .NE.0 TEMP. DEP. HEAT
C                   CAPACITY
C         16 TO 20  .EQ.0 CONSTANT HEAT GE-   IQ
C                   NERATION PER UNIT VOL.
C                   .NE.0 TEMP. DEP. Q
C
C
C....  CONDUCTIVITY IN X DIRECTION TABLE
C      THE TEMPERATURE DEPENDENT CONDUCTIVITY IN THE X
C      DIRECTION (KX) IS CALCULATED BY LINEAR INTERPOLATION
C      BETWEEN TWO CONSECUTIVE ENTRIES IN THIS TABLE. THE
C      FIRST CARD OF THIS SET OF DATA INDICATES THE NUMBER
C      OF ENTRIES IN THE TABLE. IT MUST BE FOLLOWED BY THE
C      SAME NUMBER OF CARDS WITH A PAIR OF VALUES (TEMPE-
C      RATURE AND CORRESPONDENT PROPERTY) IN EACH ONE. THEIR
C      ORDER IS CRUCIAL. THE PAIRS MUST BE ORDERED ACCORDING
C      TO THE INCREASING VALUE OF TEMPERATURE.
C      OMIT IF IKX.EQ.0
C
C      FIRST CARD -FORMAT(I5)
C
C        COLUMNS            DESCRIPTION         VARIABLE
C        1 TO 5    NUMBER OF ENTRIES OF         N
C                  THE TABLE TO BE INPUT
C
C      TABLE DATA  - FORMAT(2F10.0)
C      FOR EACH PAIR OF VALUES A CARD MUST BE INPUT. THE
C      TOTAL IS THE NUMBER SPECIFIED IN THE PRECEDING CARD.
C      LOWEST TEMPERATURE IN THE FIRST CARD, SECOND LOWEST
C      IN THE SECOND CARD, ETC.
C
C        COLUMNS            DESCRIPTION         VARIABLE
C        1 TO 10   TEMPERATURE                  XX(I)
C        11 TO 20  CONDUCTIVITY                 YY(I)
C
C
C....  CONDUCTIVITY IN Y DIRECTION TABLE
C      PREPARED IN THE SAME WAY AS THE CODUCTIVITY IN THE
C      X DIRECTION TABLE.
C      OMIT IF IKY.EQ.0
C
C
C....  HEAT CAPACITY TABLE
C      PREPARED IN THE SAME WAY AS THE CONDUCTIVITY IN THE
C      X DIRECTION TABLE.
C      OMIT IF IRCC.EQ.0
C
C
C....  HEAT GENERATION PER UNIT VOLUME TABLE
C      PREPARED IN THE SAME WAY AS THE CONDUCTIVITY IN THE
C      X DIRECTION TABLE.
C      OMIT IF IQ.EQ.0
C
C
C....  LINE BOUNDARY CONDITION DATA - FORMAT(3I5,2F10.0)
C      OMIT IF NLEC.EQ.0.
C      A CARD MUST INPUT FOR EACH LINE BOUNDARY CONDITION.
C      IF THE SAME LINE IS SUBJECTED TO MORE THAN ONE TYPE
```

88

```
C     CF BCUNCARY CCNDITICN, A CARD MUST BE USEC FCR EACH
C     CNE CF THESE TYPES.
C     THE TCTAL NUMEER OF CARCS MUST BE NLBC.
C     THE PRCPERTY VALUE IS CEFINEC:
C     FCR CCNVECTICN - CONSTANT HEAT TRANSFER COEFFICIENT
C       (IGNCREC IF KBCOND(I).EC.4)
C     FOR FLUX - FLUX PER UNIT AREA
C     FCR RADIATICN - PRODUCT CF EMISSIVITY BY STEFAN-
C                     BOLTZMAN CCNSTANT
C     THE AMEIENT TEMPERATURE IS IGNORED FOR FLUX BCUNDARY
C     CCNDITICN
C
C     CCLUMNS              DESCRIPTICN              VARIABLE
C      1 TC 5         ELEMENT NUMBER               KEL(I)
C      6 TC 1C        CODE TO INDICATE BOUN-       KBCOND(I)
C                     CARY CCNCITICN TYPE
C                     .EQ.1 FLUX
C                     .EQ.2 CCNVECTION (CONS-
C                      TANT CCEFFICIENT)
C                     .EQ.3 RACIATION
C                     .EQ.4 CCNVECTION (TEMP.
C                      DEP. CCEFF.)
C     11 TC 15        CODE TO INDICATE LINE        KLINE(I)
C                     .EQ.  1  S=+1 LINE
C                     .EQ.-1  S=-1 LINE
C                     .EQ.  2  T=+1 LINE
C                     .EQ.-2  T=-1 LINE
C     16 TC 25        PROPERTY VALUE               PRCPB(I,1)
C     26 TC 35        AMBIENT TEMPERATURE          PRCPB(I,2)
C
```
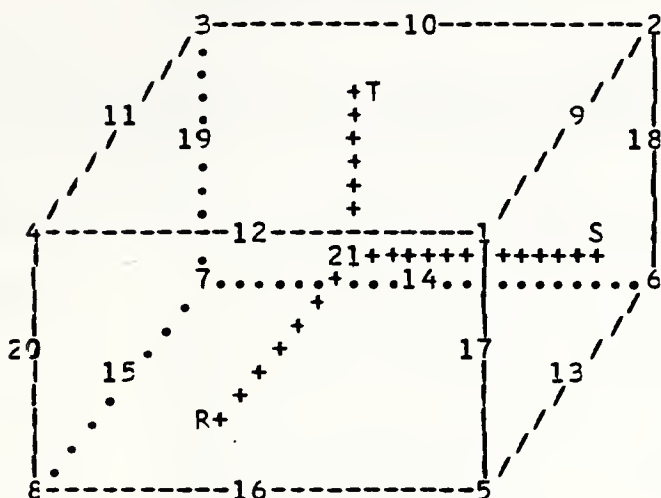
```
C
C         A.5 THREE CIMENSIONAL HEAT TRANSFER ELEMENT CATA
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C         MUST FCLLOW THE MATE MACRC CARD.
C         THE THREE CIMENSIONAL HEAT TRANSFER ELEMENT IS CALLED
C         ELMT03 AND THUS THE ELEMENT TYPE NUMBER IN CCLUMN 10
C         OF EACH MATERIAL SET NUMBER CARD MUST BE 3 WHEN THIS
C         ELEMENT IS REQUESTED. THE SECOND CARD GIVES GENERAL
C         INFCRMATION AND IS PREPARED AS FCLLOWS:
C
C
C....  GENERAL INFCRMATION DATA - FORMAT(6F10.0,4I5)
C
C         COLUMNS            DESCRIPTION            VARIABLE
C          1 TC 1C       CONDUCTIVITY IN X DIR.      D(1)
C                        (IGNORED IF TEMPERATURE
C                        DEPENDENT)
C         11 TC 2C       CONDUCTIVITY IN Y DIR.      D(2)
C                        (IGNORED IF TEMPERATURE
C                        DEPENDENT)
C         21 TC 3C       CONDUCTIVITY IN Z DIR.      D(3)
C                        (IGNORED IF TEMPERATURE
C                        DEPENDENT)
C         31 TC 40       SPECIFIC HEAT (IGNORED      D(4)
C                        IF TEMPERATURE DEPEN-
C                        DENT)
C         41 TC 50       DENSITY (IGNORED IF         D(5)
C                        TEMPERATURE DEPENDENT)
C         51 TC 6C       HEAT GENERATION PER         D(6)
C                        UNIT VCLUME (IGNORED
C                        IF TEMPERATURE DEPEN-
C                        DENT)
C         61 TC 65       NUMBER CF INTEGRATION       NGP
C                        POINTS PER DIRECTION
C                         (DEFAULT 4)
C         66 TC 7C       GEOMETRY TYPE               KAT
C                        .EQ.2 FCR AXISYMMETRY
C                        .NE.2 FCR PLANE GEOMETRY
C         71 TC 75       TCTAL NUMBER OF SURFACES    NSBC
C                        WITH SPECIFIED BOUN-
C                        DARY CCNDITICNS IN
C                        ELEMENTS WITH THE SAME
C                        MATERIAL SET NUMBER
C         76 TC 8C       CODE TC INDICATE IF ANY     INCL
C                        CF THE MATERIAL PROPER-
C                        TIES IS TEMPERATURE DE-
C                        PENDENT.
C                        .EQ.0 IF ALL MATERIAL
C                         PROPERTIES ARE CONSTANT
```

```
C                          .NE.0 IF ANY OF THE MATE-
C                          RIAL PRCPERTIES IS TEM-
C                          PERATURE DEPENDENT
C
C
C          THE FOLLOWING CATA ARE REQUIRED IF ANY OF THE MATERIAL
C          FRCPERTIES IS TEMPERATURE CEPENDENT. THEIR ORCER IS
C          CRUCIAL ANC THCSE CARCS WHICH CORRESPOND TC CCNSTANT
C          MATERIAL FFCPERTIES MUST BE OMITTED.
C
C....  MATERIAL PROPERTY CODE -FORMAT(4I5)
C          THIS CARC IS ALLWAYS REQUIRED IF INOL.NE.0 ANC MUST
C          IMMEDIATELY FCLLOW THE GENERAL INFORMATICN CATA CARC.
C
C          COLUMNS                DESCRIPTION           VARIABLE
C          1 TC 5       .EQ.0 CCNSTANT CONDUCTI-  IKX
C                       VITY IN X CIR.
C                       .NE.0 TEMP. CEP. KX
C          6 TC 1C      .EQ.0 CCNSTANT CONDUCTI-  IKY
C                       VITY IN Y DIR.
C                       .NE.0 TEMF. CEP. KY
C          11 TC 15     .EQ.0 CCNSTANT CONDUCTI-  IKZ
C                       VITY IN Z CIR.
C                       .NE.0 TEMP. CEP. KZ
C          16 TC 20     .EQ.0 CCSTANT HEAT CA     IRCC
C                       PACITY (SPECIFIC HEAT *
C                       DENSITY)
C                       .NE.0 TEMP. CEP. HEAT
C                       CAPACITY
C          21 TC 25     .EQ.0 CCNSTANT HEAT GE-   IQ
C                       NERATICN PER UNIT VOL.
C                       .NE.0 TEMP. CEP. Q
C
C
C....  CONDUCTIVITY IN X DIRECTION TABLE
C          THE TEMPERATURE DEPENDENT CONDUCTIVITY IN THE X
C          DIRECTICN (KX) IS CALCULATED BY LINEAR INTERPCLATION
C          EETWEEN TWC CCNSECUTIVE ENTRIES IN THIS TABLE. THE
C          FIRST CARC CF THIS SET CF CATA INDICATES THE NUMBER
C          CF ENTRIES IN THE TABLE. IT MUST BE FOLLCWED BY THE
C          SAME NUMEER CF CARDS WITH A PAIR OF VALUES (TEMPE-
C          RATURE ANC CCRRESPONCENT FROPERTY) IN EACH CNE. THEIR
C          CRCER IS CRLCIAL. THE PAIRS MUST BE ORDERED ACCCRDING
C          TC THE INCREASING VALUE CF TEMPERATURE.
C          CMIT IF IKX.EC.0
C
C          FIRST CARC -FCRMAT(I5)
C
C          COLUMNS                DESCRIPTION           VARIABLE
C          1 TC 5       NUMBER CF ENTRIES CF      N
C                       THE TABLE TC BE INPUT
C
C          TABLE CATA  - FORMAT(2F10.0)
C          FCR EACH PAIR OF VALUES A CARD MUST BE INPUT. THE
C          TCTAL IS THE NUMBER SFECIFIED IN THE PRECEDING CARC.
C          LOWEST TENFERATURE IN THE FIRST CARD, SECCNO LCWEST
C          IN THE SECCNC CARD, ETC.
C
C          COLUMNS                DESCRIPTICN           VARIABLE
C          1 TC 10      TEMPERATURE               XX(I)
C          11 TC 20     CONDUCTIVITY              YY(I)
C
C....  CCNDUCTIVITY IN Y DIRECTION TABLE
C          PREPAREC IN THE SAME WAY AS THE CODUCTIVITY IN THE
C          X CIRECTICN TABLE.
C          CMIT IF IKY.EC.0
C
C....  CONDUCTIVITY IN Z DIRECTION TABLE
C          FREPAREC IN THE SAME WAY AS THE CODUCTIVITY IN THE
C          X CIRECTICN TABLE.
```

91

```
C        CMIT IF IKZ.EC.0
C
C
C....  HEAT CAPACITY TABLE
C      PREPAREC IN THE SAME WAY AS THE CONDUCTIVITY IN THE
C      X CIRECTICN TABLE.
C      CMIT IF IRCC.EC.0
C
C
C....  HEAT GENERATICN PER UNIT VCLUME TABLE
C      PREPAREC IN THE SAME WAY AS THE CONDUCTIVITY IN THE
C      X DIRECTICN TABLE.
C      CMIT IF IC.EQ.C
C
C
C....  SURFACE BCLNCARY CONDITION DATA - FORMAT(3I5,2F10.0)
C      CMIT IF NSEC.EC.0.
C      A CARD MLST INPUT FCR EACH SURFACE BOUNCARY CCNDITICN.
C      IF THE SAME SURFACE IS SUBJECTED TO MORE THAN CNE TYPE
C      CF BCUNCARY CCNDITICN, A CARD MUST BE USEC FCR EACH
C      CNE OF THESE TYPES.
C      THE TCTAL NLMBER OF CARCS MUST BE NSBC.
C      THE PRCPERTY VALUE IS CEFINED:
C      FOR CONVECTICN - CONSTANT HEAT TRANSFER COEFFICIENT
C         (IGNCREC IF KBCCNC(I).EC.4)
C      FOR FLLX - FLUX PER UNIT AREA
C      FCR RACIATICN - PRODUCT OF EMISSIVITY BY STEFAN-
C                      BOLTZMAN CCNSTANT
C      THE AMBIENT TEMPERATLRE IS IGNORED FOR FLUX BCLNDARY
C      CCNDITICN
C
C         COLUMNS              DESCRIPTION            VARIABLE
C          1 TC 5         ELEMENT NUMBER              KEL(I)
C          6 TC 1C        CODE TC INCICATE BOUN-      KBCCNC(I)
C                         CARY CONCITION TYPE
C                         .EQ.1 FLUX
C                         .EQ.2 CONVECTION (CONS-
C                          TANT CCEFFICIENT)
C                         .EQ.3 RACIATION
C                         .EQ.4 CCNVECTION (TEMP.
C                          DEP. CCEFF.)
C         11 TC 15        CODE TC INCICATE SURF.      KSURF(I)
C                         .EQ.  1   R=+1  SURF
C                         .EQ.-1    R=-1  SURF
C                         .EQ.  2   S=+1  SURF
C                         .EQ.-2    S=-1  SURF
C                         .EQ.  3   T=+1  SURF
C                         .EQ.-3    T=-1  SURF
C         16 TC 25        PROPERTY VALUE              PRCPB(I,1)
C         26 TC 35        AMBIENT TEMPERATURE         PRCPB(I,2)
C
```

APPENDIX B

SAMPLE EXAMPLE

B.1 DATA CARDS

```
FEAP * * RADIAL TEMPERATURE DISTRIBUITION IN A HOLLOW CYLINDER
  33    6    1    2    1    8
COOR
   1         5    .1
  31              .3
   4         5.116666667
  29         .2833333333
   2         5    .1        3.
  32              .3        3.
   3         5    .1        6.
  33              .3        6.
   5         5.116666667    6.
  30         .2833333333    6.

POLA
   1    33    1

ELEM
   1    1    1    6    8    3    4    7    5    2    5

BOUN
   1    1   -1
   3      -1

FORC
   1    1    820.
   3         820.
```

```
MATE
   1     2              3     1
10.    6     2    10.  200.      20.
END
MACR
TANG
FORM
SOLV
DISP
END
STOP
```

94

B.2 COMPUTER OUTPUT

FEAP * * RADIAL TEMPERATURE DISTRIBUITION IN A HOLLOW CYLINDER

NUMBER OF NODAL POINTS	=	33
NUMBER OF ELEMENTS	=	6
NUMBER OF MATERIAL SETS	=	1
DIMENSION OF COORDINATE SPACE=		2
DEGREES OF FREEDOM/NODE	=	1
NODES PER ELEMENT (MAXIMUM)	=	8
EXTRA D.O.F. TO ELEMENT	=	0

FEAP * * RADIAL TEMPERATURE DISTRIBUITION IN A HOLLOW CYLINDER

NODAL COORDINATES

NODE	1 COORD	2 COORD
1	0.1000	0.0
2	0.1000	3.0000
3	0.1000	6.0000
4	0.1167	0.0
5	0.1167	6.0000
6	0.1333	0.0
7	0.1333	3.0000
8	0.1333	6.0000
9	0.1500	0.0
10	0.1500	6.0000
11	0.1667	0.0
12	0.1667	3.0000
13	0.1667	6.0000
14	0.1833	0.0
15	0.1833	6.0000
16	0.2000	0.0
17	0.2000	3.0000

18	0.2000	6.0000
19	0.2167	0.0
20	0.2167	6.0000
21	0.2333	0.0
22	0.2333	3.0000
23	0.2333	6.0000
24	0.2500	0.0
25	0.2500	6.0000
26	0.2667	0.0
27	0.2667	3.0000
28	0.2667	6.0000
29	0.2833	0.0
30	0.2833	6.0000
31	0.3000	0.0
32	0.3000	3.0000
33	0.3000	6.0000

FEAP * * RADIAL TEMPERATURE DISTRIBUITION IN A HOLLOW CYLINDER

CARTESIAN COORDINATES COMPUTED FROM POLAR INPUT WITH XO = 0.0   YO = 0.0

NODE	1-COORD	2-COORD
1	0.1000	0.0
2	0.0999	0.0052
3	0.0995	0.0105
4	0.1167	0.0
5	0.1160	0.0122
6	0.1333	0.0
7	0.1332	0.0070
8	0.1326	0.0139
9	0.1500	0.0
10	0.1492	0.0157
11	0.1667	0.0
12	0.1664	0.0087
13	0.1658	0.0174
14	0.1833	0.0

15	0.1823	0.0192
16	0.2000	0.0
17	0.1997	0.0105
18	0.1989	0.0209
19	0.2167	0.0
20	0.2155	0.0226
21	0.2333	0.0
22	0.2330	0.0122
23	0.2321	0.0244
24	0.2500	0.0
25	0.2486	0.0261
26	0.2667	0.0
27	0.2663	0.0140
28	0.2652	0.0279
29	0.2833	0.0
30	0.2818	0.0296
31	0.3000	0.0
32	0.2996	0.0157
33	0.2984	0.0314

FEAP * * RADIAL TEMPERATURE DISTRIBUITION IN A HOLLOW CYLINDER

ELEMENTS

ELEMENT	MATERIAL	1 NODE	2 NODE	3 NODE	4 NODE	5 NODE	6 NODE	7 NODE	8 NODE
1	1	1	6	8	3	4	7	5	2
2	1	6	11	13	8	9	12	10	7
3	1	11	16	18	13	14	17	15	12
4	1	16	21	23	18	19	22	20	17
5	1	21	26	28	23	24	27	25	22
6	1	26	31	33	28	29	32	30	27

FEAP * * RADIAL TEMPERATURE DISTRIBUITION IN A HOLLOW CYLINDER

NODAL B.C.

NODE   1 B.C.
1   -1
2   -1
3   -1

FEAP * * RADIAL TEMPERATURE DISTRIBUITION IN A HOLLOW CYLINDER

NODAL FORCE/DISPL

NODE   1 FORCE
1   820.0000
2   820.0000
3   820.0000

FEAP * * RADIAL TEMPERATURE DISTRIBUITION IN A HOLLOW CYLINDER

MATERIAL PROPERTIES

MATERIAL SET 1 FOR ELEMENT TYPE 2

                                    LOCAL   GLOBAL
DEGREE OF FREEDOM ASSIGNMENTS   NUMBER   NUMBER
                                       1   1

HEAT CONDUCTION ELEMENT

CONDUCTIVITY   KX =   10.00000   KY =   10.00000
SPECIFIC HEAT   0.0
DENSITY   0.0
HEAT GENER/UNIT VOL   0.0

3 GAUSS PTS/DIR
.1 LINES WITH SPECIFIED BOUNDARY CONDITIONS
PLANE   ANALYSIS

98

FEAP * * RADIAL TEMPERATURE DISTRIBUITION IN A HOLLOW CYLINDER

LINE B.C.

ELEM   B.C.   LINE   PROPERTY VALUE   TEMPERATURE
  6   2   1   200.0000   20.00000

FEAP * * RADIAL TEMPERATURE DISTRIBUITION IN A HOLLOW CYLINDER

MACRO INSTRUCTIONS

MACRO STATEMENT   VARIABLE 1   VARIABLE 2
    TANG   0.0   0.0
    FORM   0.0   0.0
    SOLV   0.0   0.0
    DISP   0.0   0.0
    END   0.0   0.0

MACRO INSTRUCTION 1 START EXECUTION TANG   V1 =   0.0   , V2 =   0.0
MACRO INSTRUCTION 2 START EXECUTION FORM   V1 =   0.0   , V2 =   0.0
FORCE CONVERGENCE TEST

  RNMAX =   5911.5   RN =   5911.5   TOL =   0.10000D-08
MACRO INSTRUCTION 3 START EXECUTION SOLV   V1 =   0.0   , V2 =   0.0
  ENERGY (DR*A*DR) = 0.47538331400 07
MACRO INSTRUCTION 4 START EXECUTION DISP   V1 =   0.0   , V2 =   0.0

FEAP * * RADIAL TEMPERATURE DISTRIBUITION IN A HOLLOW CYLINDER

PROPORTIONAL LOAD   1.0000

FEAP * * RADIAL TEMPERATURE DISTRIBUITION IN A HOLLOW CYLINDER

NODAL DISPLACEMENTS   TIME 0.0

NODE   1 COORD   2 COORD   1 DISPL
  1   0.1000   0.0   0.82000D 03

99

2	0.0999	0.0052	0.820000D 03
3	0.0995	0.0105	0.820000D 03
4	0.1167	0.0	0.722560D 03
5	0.1160	0.0122	0.722560D 03
6	0.1333	0.0	0.638110D 03
7	0.1332	0.0070	0.638110D 03
8	0.1326	0.0139	0.638110D 03
9	0.1500	0.0	0.563650D 03
10	0.1492	0.0157	0.563650D 03
11	0.1667	0.0	0.497020D 03
12	0.1664	0.0087	0.497020D 03
13	0.1658	0.0174	0.497020D 03
14	0.1833	0.0	0.436770D 03
15	0.1823	0.0192	0.436770D 03
16	0.2000	0.0	0.381750D 03
17	0.1997	0.0105	0.381750D 03
18	0.1989	0.0209	0.391750D 03
19	0.2167	0.0	0.331140D 03
20	0.2155	0.0226	0.331140D 03
21	0.2333	0.0	0.284280D 03
22	0.2330	0.0122	0.284280D 03
23	0.2321	0.0244	0.284280D 03
24	0.2500	0.0	0.240660D 03
25	0.2486	0.0261	0.240660D 03
26	0.2667	0.0	0.199850D 03
27	0.2663	0.0140	0.199850D 03
28	0.2652	0.0279	0.199850D 03
29	0.2833	0.0	0.161520D 03
30	0.2818	0.0296	0.161520D 03
31	0.3000	0.0	0.125380D 03
32	0.2996	0.0157	0.125380D 03
33	0.2984	0.0314	0.125380D 03

END OF MACRO EXECUTION

APPENDIX C

FEAP LISTING

```
      MAIN PROGRAM                                                  0000001
C.....FINITE ELEMENT ANALYSIS PROGRAM (FEAP) FOR SOLUTION OF GENERAL 0000002
C.....PROBLEM CLASSES USING THE FINITE ELEMENT METHOD. PROBLEM SIZE  0000003
C.....IS CONTROLLED BY THE DIMENSION IN BLANK COMMON AND THE VALUE OF 0000004
C.....MAX AS SET IN THE MAIN PROGRAM.                                0000005
C.....                                                               0000006
C.....PROGRAMMED BY PROF. R.L. TAYLOR, DEPARTMENT OF CIVIL ENGINEERING, 0000007
C.....UNIVERSITY OF CALIFORNIA, BERKELEY, CALIFORNIA 94720, U.S.A.   0000008
C.....                                                               0000009
C.....EXPANDED BY J.M. BETTENCOURT, DEPARTMENT OF MECHANICAL ENGINEERING, 0000010
C.....NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA 93940                  0000011
C.....                                                               0000012
C.....SET PROGRAM CAPACITY                                           0000013
      COMMON M(4000)                                                0000014
      COMMON /PSIZE/ MAX                                            0000015
      MAX = 4000                                                    0000016
      CALL PCONTR                                                   0000017
      STOP                                                          0000018
      END                                                           0000019
C                                                                   0000020
      SUBROUTINE PCONTR                                             0000021
      IMPLICIT REAL*8 (A-H,C-Z)                                     0000022
      COMMON M(1)                                                   0000023
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR      0000024
      COMMON /LABEL/ PDIS(6),A(6),BC(2),DI(6),CD(3),TE(3),FC(3)     0000025
      COMMON /CDATA/ NO1,NO,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13 0000026
      LOGICAL PCOMP                                                 0000027
      DIMENSION TITL(20),WD(3)                                      0000028
      DATA WD/4HFEAP,4HMACR,4HSTOP/                                 0000029
C.....READ A CARD AND COMPARE FIRST FOUR COLUMNS WITH MACRO LIST    0000030
1     READ(5,1000) TITL                                            0000031
      IF(PCOMP(TITL(1),WD(1))) GO TO 100                           0000032
      IF(PCOMP(TITL(1),WD(2))) GO TO 200                           0000033
      IF(PCOMP(TITL(1),WD(3))) RETURN                              0000034
      GO TO 1                                                      0000035
C.....READ AND PRINT CONTROL INFORMATION                           0000036
100   DO 101 I = 1,20                                              0000037
101   HEAD(I) = TITL(I)                                            0000038
      READ(5,1001) NUMNP,NUMEL,NUMMAT,NDM,NDF,NEN,NAD              0000039
      WRITE(6,2000) HEAD,NUMNP,NUMEL,NUMMAT,NUMMAT,NDM,NDF,NEN,NAD 0000042
C.....SET POINTERS FOR ALLOCATION OF DATA ARRAYS                   0000044
```

```
      PDIS(2) = A(NDM)
      N1 = NEN + NCF + NAD
      N01 = N1 + NUMNP*NDF*IPR
      N01 = N01 + MOD((N01-1),IPR)
      N2 = N01 + NST*2*IPR
      N3 = N2 + NEN*NDM*IPR
      N3 = N3 + NEN*IPR
      N4 = MOD((N3-1),IPR)
      N4 = N4 + NST*IPR
      N5 = N4 + NST*NST*IPR
      N6 = N5 + NUMMAT*7
      N6 = N6 + NCC((N6-1),IPR)
      N7 = N6 + 10*NUMMAT*IPR
      N8 = N7 + NCF*NUMNP
      N9 = N8 + NCD((N8-1),IPR)
      N9 = N9 + NCN*NUMNP*IPR
      N10 = N9 + NEN1*NUMEL
      N10 = N10 + MOD((N10-1),IPR)
      N11 = N10 + NDF*NUMNP*IPR
      N12 = N11 + NUMNP*NUMAP
      N13 = N12 + NDF*NUMNP
C.....  CHECK THAT SUFFICIENT MEMORY EXISTS
      CALL SETMEM(M,N13)
      CALL PNEWC(M,N13)
C.....  MESH INPUT SUBROUTINE TO READ AND PRINT ALL MESH DATA
      III = ...
      CALL PMESH(M(N2),M(N5),M(N6),M(N7),M(N8),M(N9),M(N10),M(N11),NDF,
     1   NEN1,III)
C.....  ESTABLISH PROFILE OF RESULTING EQUATIONS FOR STIFFNESS, MASS, ETC..
      CALL PROFIL(M(N12),M(N7),M(N9),NDF,NEN1,NAD)
C.....  SET POINTERS FOR SOLUTION ARRAYS * CHECK FOR SUFFICIENT MEMORY
      N13 = NEQ + NCD((N13-1),IPR)
      NE = N13 + NEQ*IPR
      CALL SETMEM(M(NE))
      CALL PNEWC(M(N13),NEQ*IPR)
      GO TO ...
C.....  MACRO SOLUTION MODULE FOR ESTABLISHING SOLUTION ALGORITHM
  200 CALL MACR(M(N01),M(NO),M(N1),M(N2),M(N3),M(N4),M(N5),M(N6),M(N7)
     1 ,M(NE),M(NE),M(N1C),M(N11),M(N12),M(N13),N,M(NE),NDF,NCN,NEN1
     2 ,M(ST,NE)
      GO TO ...
C.....  INPUT/OUTPUT FORMATS
 1000 FORMAT(2CA4)
 1001 FORMAT(16I5)
 2000 FORMAT(1H1,20A4//5X,3CHNUMBER OF NODAL POINTS      =,I6/5X,3CHNUM...
```

102

```
1BER OF ELEMENTS                   =,I6/5X,30HNUMBER OF MATERIAL SETS
2=,I6/5X,30HDIMENSION OF COORDINATE SPACE=,I6/5X,30HDEGREES OF FREE.:
3COM/NODE =,I6/5X,30HNODES PER ELEMENT (MAXIMUM) =,I6/5X,3CHE:
4XTRA C.C.F. TO ELEMENT =,I6)
      END
      BLOCK DATA
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      COMMON /LABEL/ PDIS(6),A(6),BC(2),DI(6),CD(3),TE(3),FD(3)
      DATA A/2H ,1,2H ,2H ,2H , 5,2H ,6,CD/4H COO,4HRDIN,4HATES/
      DATA TE/4H TEM,4HPERA,4HTURE/,FD/4H FOR,4HCE/D,4HISPL/
      DATA PDIS/4H(I10,2H ,,4HF13.,4H4H4,,4H6E13.,4H.5/
      DATA BC/4H B.C,2H ,,DI/4H DIS,2HPL ,4H VEL,2HOC,4H ACC,2HEL/
      DATA O/1HC/,IPR/2/
      END

      SUBROUTINE GENVEC(NDM,X,CD,PRT,ERR,PRTZ)
      IMPLICIT REAL*8 (A-H,O-Z)
C.....
C.... GENERATE REAL DATA ARRAYS BY LINEAR INTERPOLATION
C
      LOGICAL PRT,ERR,PCOMP,PRTZ
      COMMON /CDATA/O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      DIMENSION X(NDM,1),XL(7),CD(3)
      DATA BL/4H    /
      NCT = 0
      NG = 0
102   L = NG
      LGD = NG
      READ(5,1000) N,NG,XL
      IF(N.LE.0.OR.N.GT.NUMNP) GO TO 109
103   DO 103 I = 1,NDM
      X(I,N) = XL(I)
104   IF(LG) 104,102,104
      LG = ISIGN(LG,N-L)
      LI = (IABS(N-L+LG)-1)/IABS(LG)
105   DO 105 I = 1,NDM
      XL(I) = (X(I,N)-X(I,L))/LI
106   L = L + LG
      IF((N-L)*LG.LE.0) GO TO 102
      IF(L.LE.0.OR.L.GT.NUMNP) GO TO 108
107   DO 107 I = 1,NDM
      X(I,L) = X(I,L-LG) + XL(I)
108   WRITE(6,3000) L,(CD(I),I=1,3)
      ERR = .TRUE.
```

```
      GO TO 102
109   IF(.NOT.PRT) RETURN
      DO 113 J = 1,NUMNP
      IF(PRT2) GO TO 111
      DO 110 L = 1,NDM
110   IF(X(L,J).NE.0.0D0) GO TO 111
      GO TO 113
111   CONTINUE
      MCT = MCT - 1
      IF(MCT.GT.C) GO TO 112
      MCT = 50
      WRITE(6,2000) O,HEAD,((CC(L),L=1,3),(L,CD(1),CD(2),L=1,NDM)
112   IF(PCCMF(X(1,J),BL)) WRITE(6,2008) N
      IF(.NOT.FCOMP(X(1,J),BL)) WRITE(6,2009) J,(X(L,J),L=1,NDM)
113   CONTINUE
      RETURN
1000  FORMAT(2I5,7F10.0)
2000  FORMAT(A1,20A4//5X,5HNODAL,3A4//6X,4HNODE,9(I7,A4,A2))
2008  FORMAT(I10,32H HAS NCT BEEN INPUT OR GRNERATED)
2009  FORMAT(I10,9F13.4)
3000  FORMAT(5X,43H**FATAL ERRCR 02** ATTEMPT TO GENERATE NODE,I5,3H IN
     1 ,3A4)
      END

      SUBROUTINE GENELM(ICL,IX,NEN1,PRT,ERR)
      IMPLICIT REAL*8 (A-H,C-Z)
      LOGICAL PRT,ERR
      COMMON /CCATA/O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      DIMENSION ICL(1),IX(NEN1,1)

C.... READ ANC/CR GENERATE ELEMENT CCNNECTIONS
C
      L = 0
      DO 206 I = 1,NUMEL
      IF(PRT) WRITE(6,2001) O,HEAD,(K,K=1,NEN)
      J = PRINC(ALNEL,I+49)
      DO 201 N = 1,J
200   READ(5,1000) L,LK,(ICL(K),K=1,NEN),LX
      IF(L.EC.0) L = NUMEL+1
      IF(LX.EC.0) LX = 1
      IF(L-N) 201,202,203
201   WRITE(6,3001) L,N
      ERR = .TRUE.
      GO TO 206
202   NX = LX
      DO 207 K = 1,NEN
```

```
207     IF(ICL(K).GT.NUMNP.OR.ICL(K).LT.0) GO TO 208
        IX(K,L) = IDL(K)
        IX(NEN1,L) = LK
        GO TO 206
203     IX(NEN1,N) = IX(NEN1,N-1)
204     IX(K,N) = IX(K,N-1) + NX
        IX(K,N) = IX(K,N-1).EQ.0) IX(K,N) = 0
        IF(IX(K,N).GT.NUMNP.OR.IX(K,N).LT.0) GO TO 208
205     CONTINUE
        IF(PRT) WRITE(6,2002) N,IX(NEN1,N),(IX(K,N),K=1,NEN)
        GO TO 206
208     WRITE(6,2002) N
        ERR = .TRUE.
206     CONTINUE
        RETURN
1001    FORMAT(16I5)
2001    FORMAT(A1,2CA4//5X,6HELEMENTS//3X,7HELEMENT,2X,8HMATERIAL,
       1 14(I3,5H NODE))//(20X,I4(I3,5H NODE)))
2002    FORMAT(2I10,14I8/(20X,14I8))
3001    FORMAT(5X,20H**ERROR 03** ELEMENT,I5,22H APPEARS AFTER ELEMENT,I5)
3002    FORMAT(5X,20H**ERROR 04** ELEMENT,I5,18H HAS ILLEGAL NODES)
        END

        SUBROUTINE GENBCC(ICL,ID,NDF,PRT,III)
        IMPLICIT REAL*8 (A-H,O-Z)
        LOGICAL PRT
        COMMON /CCATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
        COMMON /LABEL/ PDIS(6),A(6),BC(2),DI(6),CD(3),TE(3),FD(3)
        DIMENSION IDL(6),ID(NCF,1)

C.....  READ AND/OR GENERATE RESTRAINT CODES
C
        IF(PRT) WRITE(6,2000) O,HEAD,(I,BC,I=1,NDF)
        III = 1
        NG = C
        LG = A NC
402     LG = A NC
        READ(5,1CC1) N,NG,ICL
        IF(N.LE.C.OR.N.GT.NUMNP) GO TO 60
51      II = 1,NDF
        IDL(I) = IDL(I)
        IF(L.NE.N.AND.IDL(I).EQ.C.AND.ID(I,L).LT.0) ID(I,N) = -1
        CONTINUE
        LG = ISIGN(LG,N-L)
52      L = L + LG
```

```fortran
      IF((N-L)*LG.LE.0) GO TO 402
   53 IF(IC(I,L-LG).LT.0) IC(I,L) = -1
   60 DO 58 N = 1,NUMNP
      DO 58 I = 1,NDF
      IF(IC(I,N).NE.0) GO TO 57
   56 CONTINUE
      GO TO 58
   57 IF(PRT) WRITE(6,2001) N,(ID(I,N),I=1,NDF)
   58 CONTINUE
      RETURN
 1001 FORMAT(16I5)
 2000 FORMAT(A1,2CA4//5X,17HACCAL B.C.    //6X,4HNODE,9(I7,A4,A2)/1X)
 2001 FORMAT(I10,9I13)
      END

C....
C     DATA INPUT ROUTINE FOR MESH DESCRIPTION
C
      SUBROUTINE PMESH(ICL,IE,C,ID,X,IX,F,T,NDF,NDM,NEN1,III)
      IMPLICIT REAL*8 (A-H,C-Z)
      LOGICAL PRT,ERR,PCOMP
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      COMMON /ELDATA/ DM,N,MA,MCT,IEL,NEL
      COMMON /LABEL/ PDIS(6),A(6),BC(2),DI(6),CD(3),TE(3),FD(3)
      DIMENSION IE(7,1),D(10,1),ID(NDF,1),X(NDM,1),IX(NEN1,1),XHED(7)
     1        ICL(6),F(NCF,1),T(1),WD(12),VA(2)
      DATA WD/4HCOOR,4HELEM,4HMATE,4HBOUN,4HFORC,4HTEMP,4HEND ,4HPRIN,
     1     4HNOPR,4HPAGE,4HFOLA,4HSPHE/
      DATA EL/4HELAN/,VA/4H  E/,VAL,2HUE/,LIST/12/
C....
C     INITIALIZE ARRAYS
      ERR = .FALSE.
      IF(III.LT.0) GO TO 10
      PRT = .TRUE.
  100 DO 101 N = 1,NUMNP
      DO 100 I = 1,NDF
      ID(I,N) = 0
  101 F(I,N) = 0.0D0
      X(1,N) = BLC
      T(N) = 0.0D0
   10 READ(5,1CC0) CC
      IF(FCOMP(CC,WD(I))) GO TO 30
   20 CONTINUE
   30 GO TO (1,2,3,4,5,6,7,8,9,11,12,13),I
```

106

```
C.....  NODAL COORDINATE DATA INPUT
1       CALL GENVEC(NDM,X,CC,PRT,ERR,.TRUE.)
        GO TO 1C
C.....  ELEMENT DATA INPUT
2       CALL GENELM(ICL,IX,NEN1,PRT,ERR)
        GO TO 1C
C.....  MATERIAL DATA INPUT
3       WRITE(6,2004) O,HEAD
        READ(5,1002) MA,IEL,ICL,XHEC
        IF(MA.EC.C) GO TO 1C
        DO 300 I = 1,NDF
300     IE(I,MA) = ICL(I)
        DO 301 I = 1,NDF
301     IF(ICL(I).NE.0) GO TO 303
C.....  ALL ZERO RESET TO DEFAULT VALUES
302     DO 302 I = 1,NDF
303     IE(I,MA) = IEL
304     WRITE(6,2003) MA,IEL,XHEC,(I,IE(I,MA),I=1,NCF)
        CALL ELMLIB(D(1,MA),IDL,X,IX,IDL,IDL,NCF,NDM,NDF,1)
        GO TO 1C
C.....  READ THE RESTRAINT CONDITIONS FOR EACH NODE
4       CALL GENBCC(IDL,ID,NDF,PRT,III)
        GO TO 1C
C.....  FORCE/DISPL DATA INPUT
5       CALL GENVEC(NDF,F,FC,PRT,ERR,.FALSE.)
        GO TO 1C
C.....  TEMPERATURE DATA INPUT
6       CALL GENVEC(1,T,TE,PRT,ERR,.FALSE.)
        GO TO 1C
7       IF(ERR) STOP
        RETURN
8       PRT = .TRUE.
        GO TO 1C
9       PRT = .FALSE.
        GO TO 1C
11      READ(5,1000) O
C.....  CONVERT POLAR COORDINATES TO CARTESIAN
12      CALL PCLAR (X,NDM,FFT)
        GO TO 1C
C.....  CONVERT SPHERICAL COORDINATES TO CARTESIAN
13      CALL SFEFE (X,NDM,FRT)
        GO TO 1C
1000    FORMAT(A4,75X,A1)
1002    FORMAT(IE,4X,I1,6I2,7A4)
```

107

```
2003  FORMAT(/5X,12HMATERIAL SET,I3,17H FOR ELEMENT TYPE,I2,5X,7A4//
     1    10X,45HDEGREE OF FREEDOM ASSIGNMENTS    LOCAL   GLOBAL /
     2    42X,6HNUMBER,4X,6HNUMBER/(36X,2I10))
2004  FORMAT(A1,20A4//5X,19HMATERIAL PROPERTIES)
      END

      SUBROUTINE PCLAR (X,NDM,FRT)
      IMPLICIT REAL*8 (A-H,O-Z)
C.....
C.... CONVERT POLAR TO CARTESIAN COORDINATES
C
      LOGICAL FRT
      DIMENSION X(NDM,1)
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      IF(NDM.EQ.1) RETURN
      TH = DATAN(1.0D0)/45.0D0
      READ(5,1000) NI,NE,INC,XC,YO
      IF(NI.LE.0) RETURN
      IF(NI.GT.NUMNP.OR.NE.GT.NUMNP) GO TO 300
      INC = ISIGN(MAX0(IABS(INC),1),NE-NI)
      IF(NE.EQ.0) NE = NI
      N = NI
100   R = X(1,N)
      XO = XC + R*DCOS(X(2,N)*TH)
      YO = YO + R*DSIN(X(2,N)*TH)
      MCT = MCT-1
      IF(MCT.GT.0) GO TO 250
      IF(PRT) WRITE(6,2000) O,HEAD,XO,YO,(I,I=1,NDM)
      MCT = 50
250   IF(PRT) WRITE(6,2001) N,(X(I,N),I=1,NDM)
      N = N+INC
      IF((NE-N)*INC.GE.0) GO TO 200
      IF(MOD(NE-NI,INC).EQ.0) GO TO 100
      N = NE
      GO TO 200
C....
300   ERROR
      WRITE(6,3000) NI,NE
      STOP
C....
1000  FORMAT(3I5,2F10.0)
2000  FORMAT(A1,20A4//5X,8E+CARTESIAN COORDINATES COMPUTED FROM POLAR IN
     1PUT//6X,4HNODE-XC = ,E12.4,6X YO = ,E12.4
     2 //6X,4HNODE,6(I7,6H-COORD))
```

108

```fortran
2001  FORMAT(I10,6F13.4)
3000  FORMAT(5X,4SH**FATAL ERRCR 16** ATTEMPT TO CONVERT NODES NI = ,
     1 I6,5H TO NE = ,I6)
      END

C.... MONITOR AVAILABLE MEMORY IN BLANK COMMON
C
      SUBROUTINE SETMEM(J)
C
      COMMON M(1)
      COMMON /PSIZE/ MAX
      K = J
      IF(K.LE.MAX) RETURN
      WRITE(6,1000) K,MAX
      STOP
1000  FORMAT(5X,49H**ERROR 01** INSUFFICIENT STORAGE IN BLANK COMMON/
     1 17X,11HREQUIRED =,I8/17X,11HAVAILABLE =,I8/)
      END

C.... CONVERT SPHERICAL TC CARTESIAN COORDINATES
C
      SUBROUTINE SPHERE (X,NDM,PRT)
      IMPLICIT REAL*8 (A-H,C-Z)
      LOGICAL PRT
      DIMENSION X(NDM,1)
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      IF(NDM.LT.3) RETURN
      MCT = C
      TH = DATAN(1.0D0)/45.0D0
100   READ(5,1000) NI,NE,INC,XO,YO,ZO
      IF(NI.LE.C) RETURN
      IF(NI.GT.NUMNP.OR.NE.GT.NUMNP) GO TO 300
      INC = ISIGN(MAXO(IABS(INC),1),NE-NI)
      IF(NE.EQ.C) NE = NI
200   N = NI
      ANGX = X(1,N)*TH
      ANGZ = X(3,N)*TH
      X(1,N) = XO + R*DSIN(ANGZ)*DCOS(ANGX)
      X(2,N) = YO + R*DSIN(ANGZ)*DSIN(ANGX)
      X(3,N) = ZO + DCOS(ANGZ)
      MCT = MCT - 1
      IF(MCT.GT.0) GO TO 250
      IF(PRT) WRITE(6,2000) O,HEAD,XO,YO,ZO,(I,I=1,NDM)
      MCT = 50
```

109

```
250   IF(PRT) WRITE(6,2001) N,(X(I,N),I=1,NCM)
      N = N + INC
      IF((NE-N)*INC.GE.0) GO TO 200
      IF(MOD(NE-NI,INC).EQ.0) GO TO 100
      NI = NE
      N = NE
      GO TO 200
C
C...  ERROR
300   WRITE(6,3000) NI,NE
      STOP
C
C...
1000  FORMATS
2000  FORMAT(3I5,5X,3F10.0)
2300  FORMAT(A1,20A4//5X,62HCARTESIAN COORDINATES COMPUTED FROM SPHERICA
     1L INPUT WITH XO = ,E12.4,6H YO = ,E12.4,6H ZO = ,E12.4,6H-CCCRD))
     2 /6X,4HNODE,6(I7,6H-CCCRD))
2001  FORMAT(I10,6F13.4)
3000  FORMAT(5X,49H**FATAL ERROR 16** ATTEMPT TO CONVERT NODES NI = ,
     1 I6,5H TO NE = ,I6)
      END
```

                                                      ...00429
                                                      ...00430
                                                      ...00431
                                                      ...00432
                                                      ...00433
                                                      ...00434
                                                      ...00435
                                                      ...00436
                                                      ...00437
                                                      ...00438
                                                      ...00439
                                                      ...00440
                                                      ...00441
                                                      ...00442
                                                      ...00443
                                                      ...00444
                                                      ...00445
                                                      ...00446
                                                      ...00447
                                                      ...00448
                                                      ...00449

```fortran
      SUBROUTINE PMACIO(CT,WD,ENDM,ERR,IPR,LL,NE)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C.... MACRO PROGRAM INPUT/OUTPUT ROUTINE
C
      LOGICAL ERR,PCOMP
      DIMENSION CT(4,1),WD(1)
C.... READ MACRO CARDS
      LMAX = 16
      CALL SETMEM(NE+LMAX*4*IPR)
      CT(1,1) = WD(7)
      CT(3,1) = 1.0
      LL = 1
100   LL = LL + 1
      IF(LL.LT.LMAX) GO TO 110
      LMAX = LMAX + 16
      CALL SETMEM(NE+LMAX*4*IPR)
      READ(5,1000) (CT(J,LL),J=1,4)
      WRITE(6,2000) (CT(J,LL),J=1,4)
      IF(.NOT.PCOMP(CT(1,LL),ENDM)) GO TO 100
110   CT(1,LL) = WD(8)
C.... SET LOOP MARKERS
      NE = NE + LMAX*4*IPR
      LX = LL-1
      DO 230 L = 1,LX
      IF(.NOT.PCOMP(CT(1,L),WD(7))) GO TO 230
      JK = J + 1
      DO 210 I = K,LL
      IF(PCOMP(CT(1,I),WD(7))) J = J + 1
      IF(J.GT.9) GO TO 401
      IF(PCOMP(CT(1,I),WD(8))) J = J - 1
      IF(J.EQ.0) GO TO 220
210   CONTINUE
      GO TO 400
220   CT(4,I) = L
      CT(4,L) = I
230   CONTINUE
      J = 0
      DO 240 L = 1,LL
      IF(PCOMP(CT(1,L),WD(7))) J = J+1
      IF(PCOMP(CT(1,L),WD(8))) J = J-1
240   IF(J.EQ.0) RETURN
      WRITE(6,4000)
      ERR = .TRUE.
      RETURN
401   WRITE(6,4001)
      ERR = .TRUE.
```

```
      RETURN
1000  FORMAT(A4,1X,A4,1X,2F10.0)                                         00498
2000  FORMAT(10X,A4,1X,A4,1X,2G15.5)                                     00499
4000  FORMAT(5X,46H***FATAL ERROR 10** UNBALANCED LOOP/NEXT MACROS )     00500
4001  FORMAT(5X,45H***FATAL ERROR 11** LOOPS NESTED DEEPER THAN 8)       00501
      END                                                                00502
C....                                                                    00503
      SUBROUTINE PMACR (UL,XL,TL,LD,P,S,IE,D,ID,X,IX,F,T,JDIAG,B,DR,CT   00504
     1,NDF,NEN,NEN1,NST,NEQ)                                             00505
      IMPLICIT REAL*8 (A-H,O-Z)                                          00506
C....                                                                    00507
C.... MACRO INSTRUCTION SUBPROGRAM                                       00508
C....                                                                    00509
C.... CONTROLS PROBLEM SOLUTION AND OUTPUT ALGORITHMS BY                 00510
C.... ORDER OF SPECIFYING MACRO COMMANDS IN ARRAY WD.                    00511
C                                                                        00512
      LOGICAL AFR,BFR,CFR,AFL,EFL,CFL,DFL,EFL,FFL,GFL,TFL,PCOMP          00513
      LOGICAL ERR,EER                                                    00514
      COMMON A(1)                                                        00515
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR           00516
      COMMON /EAGYS/ AENGY                                               00517
      COMMON /LABEL/ NDIS(6),Z(6),BC(2),DI(6),CD(3),TE(3),FD(3)          00518
      COMMON /MDATA/ NO1,NC,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13   00519
      COMMON /PRLOD/ PROP                                                00520
      COMMON /FSIZE/ MAX                                                 00521
      COMMON /TDATA/ TIME,DT,C1,C2,C3,C4,C5                              00522
      DIMENSION WD(25),CT(4,1),CTL(4,1),LVE(5),JDIAG(1)                  00523
     1 UL(1),XL(1),TL(1),LD(1),P(1),S(1),LVS(9),JDIAG(1),                00524
     2 IX(1),F(1),T(1),B(1),LC(1),P(1),IE(7,1),D(1),ID(1),X(1),          00525
      DATA WD/4HTOL ,4HDT  ,4HCL  ,4HDATA,4HSTRE,4HDISP,4HFORM,4HLOOP,4HNEXT, 00526
     1 4HPROP,4HDATA,4HTIME,4HCONV,4HSCLV,4HLMAS,4HMESH,                 00527
     2 4HEIGE,4HEXCC,4HUTAN,4HREAC            ,4HWRITE,                  00528
     4 4HCHEC,                                 4HOCEI,4HUSER,            00529
      DATA NWC/25/,ENDM/4HEND /,NV/1/,NC/1/,NA/1/                        00530
      INTEGER CLCCK(6)                                                   00531
C.... SET INITIAL VALUES OF PARAMETERS                                   00532
      CT =0.0CC                                                          00533
      PROP =1.0 CC                                                       00534
      RNMAX =0.0CC                                                       00535
      TIME =0.0CC                                                        00536
      TOL =1.0CC                                                         00537
      UN =0.0CO                                                          00538
      AFR =.TRUE.                                                        00539
      AFL =.FALSE.                                                       00540
      BFL =.TRUE.                                                        00541
      BFR =.FALSE.                                                       00542
```

112

```fortran
      CFL = .TRUE.
      CFR = .FALSE.
      CEFL = .TRUE.
      EEFL = .FALSE.
      FFL = .TRUE.
      CFFL = .FALSE.
      TFL = .FALSE.
      NE = NENC
      NNEG = NCF*NUMNP
      NPLD = 0
      WRITE(6,2001) O,HEAC
C.....READ MACRC CARDS
      ERR = .FALSE.
      CALL FRACIC(CT,WD,ENCM,ERR,IPR,LL,NE)
      IF(ERR) RETURN
C.....EXECUTE MACRO INSTRUCTION PROGRAM
      LV = 0
      L = L - 1
299   DO 300 J = 1,NWD
      IF(PCOMP(CT(1,L),WD(J))) GO TO 310
300   CONTINUE
      WRITE(6,3000) I
310   CONTINUE
      IF(L.EQ.1.OR.L.EQ.LL) GO TO 311
      WRITE(6,2010) I,(CT(K,L),K=1,4)
      IF(.NOT.TFL) GO TO 311
311   CALL SETIME(ITIME(0)*0.01C0)
      LCTM = L
      LLL = L
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,
     1       27,             21,32,33),J
C.....SET SCLUTICN TOLERANCE
1     TOL = CT(3,L)
      GO TO 300
C.....SET TIME INCREMENT
2     CT = CT(3,L)
      GO TO 300
C.....PRINT STRESS VALUES
3     LX = CT(3,LX)
      IF(DMCL.LVE(LV))DMAX1(CT(3,LX),DMAX1(CT(3,L),1.0D0)),EG.0.0C0)
     1   CALL FFCRM(UL,XL,TL,LD,P,S,IE,D,ID,X,IX,F,T,JDIAG,DR,DR,.FALSE.,,1)
      NCF = NDM,NEN1,NST,4,B,N(NV),.FALSE.,.FALSE.,.FALSE.,1)
      CT = CT(3,L)
      GO TO 300
C.....PRINT CISPLACEMENTS
4     LX = CT(3,LX)
      IF(DMCL(CT(3,LX),DMAX1(CT(3,L),1.0D0)).NE.C.CD0) GO TO 339
```

113

```
      WRITE(6,2002) O,HEAD,TIME,PRCP
      CALL PRTDIS(IC,X,B,F,NDM,NDF)
      GO TO 330
C.... FORM TANGENT STIFFNESS
19    IF(CFL)CALL PSETM(NC,NE,JDIAG(NEQ)*IPR,CFL)
      CFR = .TRUE.
5     IF(J.EQ.5)CFR = .FALSE.
      IF(GFL)CALL PSETM(NA,NE,JDIAG(NEQ)*IPR,GFL)
      IF(NFL.GT.O) PROP = PROPLD(TIME,0)
      CALL PZERC(M(NA),JCIAG(NEQ)*IPR)
      CALL PFORM(UL,XL,TL,LD,P,S,IE,D,ID,X,IX,F,T,JDIAG,DR,M(NA),M(NC),
     ?          NCF,NEN1,NST,3,B,M(NV),.TRUE.,.FALSE.,,CFR,.FALSE.,,1)
2     AFR = .TRUE.
      GO TO 330
C.... FORM OUT OF BALANCE FORCE FOR TIME STEP/ITERATION
6     IF(NFLD.GT.O)PROP = PRCFLD(TIME,0)
      CALL PLOAD(IC,F,DR,NNEQ,PROP)
      CALL PFORM(UL,XL,TL,LC,P,S,IE,D,ID,X,IX,F,T,JDIAG,DR,M(NA),M(NC),
     ?          NCF,NEN1,NST,6,B,M(NV),.FALSE.,.TRUE.,,FALSE.,,1)
2     EFR = .TRUE.
      RN = DSQRT(DOT(DR,DR,NEC))
      RNMAX = CMAX1(RNMAX,RN)
      WRITE(6,2005) RNMAX,RN,TCL
      IF(RN.GE.RNMAX*TOL) GO TO 330
      LX = LV
      LO = LLVS(LV)
      CT(3,LX) = CT(3,LO)
      GO TO 330
C.... SET LOOP START INDICATORS
7     LV = LV + 1
      LX = CT(4,L)
      LLVS(LV) = LX
      LVE(LV) = L
      GO TO 330
C.... LOOP TERMINATOR CONTROL
8     N = CT(3,L)
      CT(3,L) = CT(3,L) + 1.0DO
      IF(CCT(3,L).GT.CT(3,N)) LV = LV -1
      IF(CCT(3,L).LE.CT(3,N)) L = N -1
      GO TO 330
C.... INPUT PROPORTIONAL LOAD TABLE
9     NPLD = CT(3,L)
      PRCP = PRCPLD(0.0DO,NFLD)
      GO TO 330
C.... READ COMMAND
```

114

```
10      READ(5,1000) (CTL(I),I=1,4)
        IF(.NOT.PCOMF(CT(2,L),WE(1),CTL(1))) GO TO 402
        IF(PCOMF(CTL(1),WE(2))) TCL = CTL(3)
        IF(PCOMF(CTL(1),WE(4))) CT = CTL(3)
        GO TO ...                         GO TO 101
        ISW = CT(3,L)
        IF(ISW.NE.0) ISW = 5
        IF(ISW.NE.5) REWIND ISW
        READ(ISW) TIME,(B(I),I=1,NEQ)
        GO TO ...
101     INCREMENT TIME
        TIME = TIME + DT
        RNMAX = 0.0D0
        UN = 0.0D0
        GO TO TC
C....
11      COMPUTE CONVERGENCE TESTS
C....
12      RN = CCT(CR,CR,NEQ)
        LN = DCT(B,B,NEQ)
        UN = DMAX1(UN,RN)
        CN = DC(CRT(RN))
        RN = CRT(RN)               CN,RN,TOL
        WRITE(6,2002)
        LXO = LL=LL(LL))
        IF(RN.LT.CN*TOL) CT(3,LX) = CT(3,LO)
        GO TO TC
C....
13      SOLVE THE EQUATIONS
        GO TO TC  131
        IF(CCFR) CALL ACTCOL(M(NA),DR,JDIAG,NEG,AFR,BFR)
131     IF(BFR) WRITE(6,2007) AENGY
132     GO TO 132
        CALL UACTCL(M(NA),M(NC),CR,JDIAG,NEQ,AFR,BFR)
        AFR = .NOT.BFR) GO TO 330
        IF(.NOT.BFR) GO TO ...
        DO 133 N=1,NEQ
133     B(N) = B(N) + DR(N)
        GO TO TC
C....
14      FORM A LUMPED MASS APPROXIMATION
        AFL = .FALSE.
        EFL = .TRUE.
        IF(EFL) CALL PSETM(AN,NE,NEG*IPR,EFL)
139     CALL FZFC (M(NN),NEG*IPR)
        GO TO TC
C....
15      FORM A CONSISTENT MASS APPROXIMATION
        AFL = .TRUE.
        BFL = .FALSE.
```

```
152       IF(DFL) CALL PSETM(NM,NE,JDIAG(NEQ)*IPR,DFL)
140       CALL PZERO(M(NM),JDIAG(NEC)*IPR)
        1 N(AN),ACF,ACM,NEN1,NST,5,B,M(NV),AFL,BFL,.FALSE.,4)
          GO TO 200
C....
16        CHANGE REST CATA
          I=-1
          CALL PREST(LD,IE,D,ID,X,IX,F,T,NDF,NDM,NEN1,1)
          IF(I.GT.0) GO TO 404
          GO TO 200
C....
17        COMPUTE DOMINANT EIGENVALUE BY INVERSE ITERATION
          J = NN
          CALL PEIGS(M(NA),M(J),F,X,B,DR,ID,IX,JDIAG,NDF,NDM,NEN1,CFL)
          IF(DFL) GO TO 404
          GO TO 200
18        GO TO 181
C....
          MACRO *EXEC* EXPLICIT INTEGRATION OF EQUATIONS OF MOTION
          NQ = MOD((NQ-1),IPR)
          NQ = NQ + NEC*IPR
          NV = NV + NDF*NUMNP*IFR
          NE = NE + NEC*IPR
          CALL SETMEM(NE,NE-NC)
          CALL PZERO(M(NQ),NE-NC)
          FFL = .TRUE.
          GO TO 200
181       IF(.NOT.BFR.OR.EFL) GO TO 403
          CALL ELPLOT(DR,B,M(NC),M(NR),M(NN),DT,NEQ)
          GO TO 200
C....
20        COMPUTE REACTIONS AND PRINT
          CALL PZERO(DR,NNEG*IFR)
          CALL REFLC(DR,UL,XL,TL,LD,P,S,IE,D,ID,X,IX,F,T,JDIAG,DR,CR,CR,
        1 NEN1,NST,6,B,M(NV),.FALSE.,.TRUE.,.FALSE.,.TRUE.,1)
          CALL PRTREA(DR,NDF)
          GO TO 200
C....
23        PRINT ARRAYS AS OUTPUT
          IF(PCOMP(CT(2,L),WD(5)).OR.CR.PCCMP(CT(2,L),WC(15)))
        1 WRITE(6,2006) (N,JDIAG(N),N=1,NEQ)
          IF(PCCNF(CT(2,L),WC(5)).AND.AFR) CALL PRITE(M(NA),JDIAG(NEQ),
        1 SHSTIFFNES)
          IF(PCCMP(CT(2,L),WD(6)).AND.BFR) CALL PRITE(CR,NEQ,
        1 SHFORCE)
          IF(PCOMP(CT(2,L),WD(15)).AND.AFL) CALL PRITE(M(NM),JDIAG(NEQ),
        1 15LCASSISTENT MASS)
          IF(PCCMP(CT(2,L),WD(14)).AND.BFL) CALL PRITE(M(NN),NEQ,
        1 11HLLUMPEC MASS)
          GO TO 200
C.....
          CHECK MESH FOR ERRORS
```

```fortran
27    CALL FFORM(UL,XL,TL,LC,P,S,IE,D,ID,X,IX,F,T,JDIAG,DR,CR,
     1 NDF,NDM,NEN1,NST,2,B,M(NV),.FALSE.,.FALSE.,1)
      GO TO 330
C.... FIRST ORDER ODE SOLUTION
31    CALL PCCE1(CT(2,L),R(NA),M(NC),M(NM),M(NN),E,DR,JDIAG,ID,UL,
     1 NDF,(N12-NOI),NDF,AFR,CFR,AFL,BFL,EER)
      IF(EER) WRITE(6,30CC) {
      GO TO ...30
C.... USER SUPPLIEC SUBROUTINE
32    STOP
C....
33    SET TIME COUNTER
      TFL = .TRUE.
      GO TO ...
330   L=...
      IF(LLL.EC.1.OR.LLL.EC.LL) GO TO 331
      IF(.NCT.TFL.) GO TO ...31
      CALL GETIME(IET)
      WCTM = IITIME(0)*0.000026D0 - WCTM
      CPUTM = IET
      WRITE(6,2015) I,(CT(K,LLL),K=1,4),WCTM,CPUTM
331   CONTINLE
      IF(L.LE.LL) GO TO 255
      WRITE(6,2CC8)
      RETURN
402   WRITE(6,4C02)
      RETURN
403   WRITE(6,4C03)
404   WRITE(6,4C04)
      RETURN
1000  FORMAT(A4,1X,A4,1X,2F10.0)
2001  FORMAT(A1,20A4//5X,18HMACRO INSTRUCTIONS//5X,15HMACRO STATEMENT,5X,
     1 10HVARIABLE 1,5X,1CTVARIABLE 2)
2002  FORMAT(5X,29HDISPLACEMENT CONVERGENCE TEST/10X,7HUNMAX =,G15.5,5X,
     1 7HUN =,G15.5,5X,7HTOL =,G15.5)
2003  FORMAT(A1,20A4,10X,4HTIME =,G13.5//5X,17HPROPORTIONAL LOAD,G13.5)
2004  FORMAT(5X,4HCN =,G12.5,5X,4HDN =,G12.5,5X,4HUN =,G12.5,5X,4HAG =
2005  FORMAT(5X,22HFORCE CONVERGENCE TEST/10X,7HRNMAX =,G15.5,5X,
     1 7HRN =,G15.5)
2006  FORMAT(//10X,2HDIAGONAL ADDRESS ARRAY /(1X,2CI5))
2007  FORMAT(5X,18HENERGY CF MACRO INSTRUCTION =,E18.10)
2008  FORMAT(5X,26H***END OF MACRO INSTRUCTION**)
2010  FORMAT(2X,19H***MACRO INSTRUCTION,I4,20H START EXECUTION**  ,2(A4,2,
     1 1X), 6H V2 =,G13.4/)
2015  FORMAT(2X,19H***MACRO INSTRUCTION,I4,13H EXECUTED**  ,2(A4,2X), 6H
     1 V1 =,G13.4/ 8H VE =,G13.4,
     2 25X,2CH WALL CLOCK TIME = ,F7.2,7HSECONDS,
     3 35X,13H CPU TIME = ,F7.2,7HSECONDS)
```

```
3000 FORMAT(5X,25H**WARNING** MACRO INSTRUCTION ,I4,             ...007786
     1 18H IS NOT EXECUTABLE)                                    ...007787
4002 FORMAT(5X,57H**FATAL ERROR 12** MACRO LABEL MISMATCH ON A READ COM ...007788
     1MAND)                                                      ...007789
4003 FORMAT(5X,63H**FATAL ERROR 13** MACRO EXCD MUST BE PRECEDED BY LMA ...007790
     1S AND FCRM)                                                ...007791
4004 FORMAT(5X,84H**FATAL ERROR 14** ATTEMPT TO CHANGE BOUNDARY RESTRAI ...007792
     1NT CCCES DURING MACRO EXECUTION )                          ...007793
     END                                                        ...007794
```

118

```
      SUBROUTINE ACTCOL(A,B,JDIAG,NEQ,AFAC,BACK)
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL AFAC,BACK
      COMMON/ENGYS/AENGY
      DIMENSION A(1),B(1),JDIAG(1)
      DATA TOL/1.0-7/
C.....
C.....  ACTIVE COLUMN PROFILE SYMMETRIC EQUATION SOLVER
C.....
C.....  FACTOR A TO UT*D*U, REDUCE B
      AENGY = 0.0D0
      JR = 0
      DO 600 J = 1,NEQ
      JD = JDIAG(J)
      JH = JD - JR
      IS = J - JH + 2
      CG = A(JD)
      IF(.NOT.AFAC) GO TO 500
      IE = J - 1
      K = JR + 2
      ID = JDIAG(IS-1)
C.....  REDUCE ALL EQUATIONS EXCEPT DIAGONAL
      DO 200 I = IS,IE
      IR = ID
      ID = JDIAG(I)
      IH = MIN0(ID-IR-1,I-IS+1)
      IF(IH.GT.0) A(K) = A(K) - DOT(A(K-IH),A(ID-IH),IH)
200   K = K + 1
C.....  REDUCE DIAGONAL TERM
300   IF(.NOT.AFAC) GO TO 500
      IR = JR + 1
      IE = JD - 1
      K = J - JD
      DO 400 I = IR,IE
      ID = JDIAG(K+I)
      D = A(I)/A(ID)
      A(I) = A(I) - D*A(ID)
400   A(JD) = A(JD) - D*A(I)
      CONTINUE
      IF(DG*A(JD).LT.0.0D0) WRITE(6,2000) J
      IF(DABS(A(JD)).LT.TOL*DABS(DG)) WRITE(6,2001) J
      IF(A(JD).EQ.0.0D0) WRITE(6,2002) J
C.....  REDUCE RHS
500   IF(BACK) B(J) = B(J) - DOT(A(JR+1),B(IS-1),JH-1)
550   IF(A(JD).NE.0.0D0.AND.AFAC) A(JD) = 1.0D0/A(JD)
600   JR = JD
      IF(.NOT.BACK) RETURN
```

```
C..... DIVIDE BY DIAGONAL PIVOTS
      DO 700 I = 1,NEQ
      ID = JDIAG(I)
      AENGY = AENGY + B(I)*B(I)*A(ID)
700   B(I) = B(I)*A(ID)
C..... BACKSUBSTITUTE
800   JD = JDIAG(J)
      C = B(J)
      JR = JDIAG(J)
      IF(J.LE.C) RETURN
      IF(JC-JR.LE.1) GO TC 1000
      IS = JD + JR + 2
      K = JR - IS + 1
      DO 900 I = IS,J
      B(I) = B(I) - A(I+K)*D
900   JC = JR
1000  GO TC 800
2000  FORMAT(5X,31H **WARNING C1** SIGN OF EQUATION,I5,18H CHANGED IN ACT..
     1CCL)
2001  FORMAT(5X,23H***WARNING,02** EQUATION,I5,45H LCST AT LEAST 7 SIGNIF..
     1ICANT DIGITS IN ACTCCL)
2002  FORMAT(5X,32H **WARNING 02** PIVOT OF EQUATICN,I5,18H IS ZERO IN AC..
     1TCCL)
      END

      SUBROUTINE ADDSTF(A,B,C,S,P,JDIAG,LD,NST,NEL,AFL,BFL,CFL,ITP)
C..... ASSEMBLE GLOBAL ARRAYS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL AFL,BFL,CFL
      DIMENSION A(1),B(1),JDIAG(1),P(1),S(NST,1),LC(1) ,C(1)
      DO 200 J = 1,NEL
      K = IAES(LC(J))
      IF(K.EG.0) GO TO 200
      IF(BFL) B(K) = B(K) + P(J)
      IF(.NOT.AFL.AND..NOT.CFL) GO TO 200
      LD = JDIAG(K) - K
      DO 100 I = 1,NEL
      M = IABS(LC(I))
      IF(M.GT.K.CR.M.EQ.0) GO TO 100
      IF(AFL) A(M) = A(M) + S(I,J)
      IF(CFL) C(M) = C(M) + S(J,I)
100   CONTINUE
```

120

```
008891  200   CONTINUE
008892        RETURN
008893        END
008894  C
008895  C.... VECTOR DOT PRODUCT
008896  C
008897        FUNCTION DOT(A,B,N)
008898        IMPLICIT REAL*8 (A-H,O-Z)
008899        DIMENSION A(1),B(1)
008900        DOT = 0.0D0
008901        DO 100 I = 1,N
008902  100   DOT = DOT + A(I)*B(I)
008903        RETURN
008904        END
008905        LOGICAL FUNCTION PCOMP(A,B)
008906        IMPLICIT REAL*8 (A-H,O-Z)
008907        PCOMP = .FALSE.
008908  C.... PCOMP MAY BE NECESSARY TO REPLACE THE FOLLOWING ALPHANUMERIC
008909  C.... COMPARISON STATEMENT IF COMPUTER PRODUCES AN OVERFLOW.
008910        IF(A.EQ.B) PCOMP = .TRUE.
008911        RETURN
008912        END
008913        SUBROUTINE ELMLIB(D,U,X,IX,T,S,P,I,J,K,ISW)
008914        IMPLICIT REAL*8 (A-H,O-Z)
008915  C
008916  C.... ELEMENT LIBRARY
008917  C
008918        COMMON /ELDATA/ DM,N,MA,MCT,IEL,NEL
008919        DIMENSION P(K),S(K,K),D(1),U(1),X(1),IX(1),T(1)
008920        IF(IEL.LE.0.OR.IEL.GT.5) GO TO 400
008921        IF(ISW.LT.3) GO TO 30
008922        DO 20 L = 1,K
008923        P(L) = 0.0D0
008924        DO 20 N = 1,K
008925  20    S(L,N) = 0.0D0
008926  30    GO TO (1,2,3,4,5),IEL
008927  1     CALL ELMT01(D,U,X,IX,T,S,P,I,J,K,ISW)
008928        GO TO 10
008929  2     CALL ELMT02(D,U,X,IX,T,S,P,I,J,K,ISW)
008930        GO TO 10
008931  3     CALL ELMT03(D,U,X,IX,T,S,P,I,J,K,ISW)
008932        GO TO 10
008933  4     CALL ELMT04(C,U,X,IX,T,S,P,I,J,K,ISW)
008934        GO TO 10
```

121

```
          CALL ELMTC5(D,U,X,IX,T,S,P,I,J,K,ISW)                        ::00939
 5        RETURN                                                       ::00940
 100      WRITE(6,4000) IEL                                            ::00941
          STOP                                                         ::00942
 4000     FORMAT(5X,39H**FATAL ERROR 04** ELEMENT CLASS NUMBER,I3,6H INPUT)  ::00943
          END                                                          ::00944
                                                                       ::00945
          SUBROUTINE EUPDAT(DR,U,V,A,XM,DT,NEQ)                        ::00946
          IMPLICIT REAL*8 (A-H,O-Z)                                    ::00947
          DIMENSION U(1),V(1),A(1),DR(1),XM(1)                         ::00948
 C....    UPDATE SOLUTION USING EXPLICIT CENTRAL DIFFERENCES           ::00949
          DATA C1+P/C.0D0/                                             ::00950
          DTH = DT/2.0D0                                               ::00951
          DTH+P = DTH + DTHP                                           ::00952
          DT+P = DTH +                                                 ::00953
          DO 100 N = 1,NEQ                                             ::00954
          A(N) = DR(N)/XM(N)                                           ::00955
          V(N) = V(N) + DTAV*A(N)                                      ::00956
 100      U(N) = U(N) + DT*V(N)                                        ::00957
          RETURN                                                       ::00958
          END                                                          ::00959
                                                                       ::00960
          SUBROUTINE NORM(X,Y,N)                                       ::00961
          IMPLICIT REAL*8 (A-H,O-Z)                                    ::00962
 C....                                                                 ::00963
 C        NORMALIZE Y TO UNIT VECTOR X                                 ::00964
          DIMENSION X(1),Y(1)                                          ::00965
          SCALE = DSQRT(DOT(Y,Y,N))                                    ::00966
          DO 100 I = 1,N                                               ::00967
 100      X(I) = Y(I)/SCALE                                            ::00968
          RETURN                                                       ::00969
          END                                                          ::00970
                                                                       ::00971
          SUBROUTINE PADDM(A,B,C,N)                                    ::00972
          IMPLICIT REAL*8 (A-H,O-Z)                                    ::00973
 C....                                                                 ::00974
 C        ADD LINEAR COMBINATION OF TWO ARRAYS TOGETHER                ::00975
          DIMENSION A(1),B(1)                                          ::00976
          DO 100 N=1,NN                                                ::00977
 100      A(N)= A(N) + C*B(N)                                          ::00978
          RETURN                                                       ::00979
          END                                                          ::00980
```

122

```
      SUBROUTINE PACDV(A,B,ID,C,NN)
      IMPLICIT REAL*8 (A-F,C-Z)
      DIMENSICN A(1),B(1),NN
      CO 100 N=1,NN
      K = ID(N)
      IF(K.GT.C) A(N) = A(N) + C*B(K)
  100 CCNTINUE
      RETURN
      END
C
C.... ADD LINEAR CCMBINATICN OF DIAGONAL ARRAY TO CIAGONAL OF AN ARRAY
C
      SUBROUTINE PADDD(A,B,JD,C,NN)
      IMPLICIT REAL*8 (A-F,C-Z)
      DIMENSICN A(1),B(1),JD(1)
      CO 100 N=1,NN
      J = JD(N)
      A(J) = A(J) + C*B(N)
  100 CONTINUE
      RETURN
      END
C
C.... SCALE AN ARRAY BY A CCNSTANT
C
      SUBROUTINE PSCAL (A,C,NN)
      IMPLICIT REAL*8 (A-F,C-Z)
      DIMENSICN A(1)
      CO 100 N=1,NN
      A(N) = C*A(N)
  100 CONTINUE
      RETURN
      END
C
C.... ROUTINE TC FORM C = C + A*B WHERE A IS A DIAGONAL MATRIX,
C.... B AND C ARE VECTORS
C
      SUBROUTINE PRODIA (A,B,C,NEC)
      IMPLICIT REAL*8 (A-F,C-Z)
      DIMENSICN A(1),B(1),C(1)
      CO 10 N=1,NEQ
   10 C(N) = C(N) + A(N)*E(N)
      RETURN
      END
```

```
      SUBROUTINE PROMUL(A,B,C,JDIAG,NEQ)                              ::::01035
      IMPLICIT REAL*8 (A-H,O-Z)                                       ::::01036
      DIMENSION A(1),B(1),C(1),JDIAG(1)                               ::::01037
C.....                                                                ::::01038
C.....ROUTINE TO FORM C = C + A*B WHERE A IS A SYMMETRIC SQUARE MATRIX ::::01039
C.....STORED IN PROFILE FCRM,B,C ARE VECTORS, AND JDIAG LOCATES THE   ::::01040
C     DIAGONALS IN A                                                  ::::01041
                                                                      ::::01042
      JS=1                                                            ::::01043
      DO 200 J=1,NEQ                                                  ::::01044
      JD=JDIAG(J)                                                     ::::01045
      IF(JS.GT.JD) GO TO 200                                          ::::01046
      EJ=B(J)*JD                                                      ::::01047
      AB=A(JD)*BJ                                                     ::::01048
      IF(JS.EC.JD) GO TO 150                                          ::::01049
      JB=J-JC                                                         ::::01050
      JE=JC-1                                                         ::::01051
      DO 100 JJ=JS,JE                                                 ::::01052
      AB=AB+ A(JJ)*B(JJ+JB)                                           ::::01053
  100 C(JJ+JB)=C(JJ+JB) + A(JJ)*BJ                                    ::::01054
  150 C(J)=C(J) + AB                                                  ::::01055
  200 JS=JD+1                                                         ::::01056
      RETURN                                                          ::::01057
      END                                                             ::::01058
                                                                      ::::01059
                                                                      ::::01060
      FUNCTION FRCPLD(T,J)                                            ::::01061
      IMPLICIT REAL*8 (A-H,O-Z)                                       ::::01062
C.....                                                                ::::01063
C.....PROPORTICNAL LOAD TABLE (CNE LOAD LOAD CARD CNLY)               ::::01064
C                                                                     ::::01065
      DIMENSION A(5)                                                  ::::01066
      IF(J.GT.C) GO TO 20C                                            ::::01067
C.....COMPLTE VALUE AT TIME T                                         ::::01068
      FRCPLC = 0.0C0                                                  ::::01069
      IF(T.LT.TMIN.OR.T.GT.TMAX) RETURN                               ::::01070
      L=MAX0(L,1)                                                     ::::01071
      PROPLD = A(1) + A(2)*T + A(3)*(DSIN(A(4)*T+A(5)))**L            ::::01072
      RETURN                                                          ::::01073
C.....INPUT TABLE CF PROPCRTICNAL LOADS                               ::::01074
  200 I=I+1                                                           ::::01075
      READ(5,1000) K,L,TMIN,TMAX,A                                    ::::01076
      WRITE(6,2000) I,K,L,TMIN,TMAX,A                                 ::::01077
      RETURN                                                          ::::01078
 1000 FORMAT(2I5,7F10.0)                                              ::::01079
 2000 FORMAT(5X,23HPROPORTICNAL LOAD TABLE//24H    NUMBER    TYPE    EXP., ::::01080
```

```
      1 14H MINIMUM TIME,15H  MAXIMUM TIME,  5X,2HA1,13X,2HA2,13X,2HA3,    ...01083
      2 12X,2HA4,13X,2HA5/(3I8,7G15.5))                                   ...01084
        END                                                              ...01085
                                                                         ...01086
        SUBROUTINE PRTDIS(IC,X,B,F,NDM,NDF)                              ...01087
        IMPLICIT REAL*8 (A-H,O-Z)                                        ...01088
C                                                                        ...01089
C.... OUTPUT ACCEL VALUES                                                ...01090
C                                                                        ...01091
        LOGICAL FCOMP                                                    ...01092
        COMMON/FFLCD/ PROP                                               ...01093
        COMMON/CCDATA/O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR          ...01094
        COMMON/LABEL/ PDIS(6),A(6),BC(2),DI(6),CD(3),TE(3),FD(3)         ...01095
        COMMON/TCDATA/ TIME,CT,C1,C2,C3,C4,C5                            ...01096
        DIMENSION X(NDM,1),B(1),LL(6),ID(NDF,1),F(NDF,1)                 ...01097
        DATA BL/4HBLAN/                                                  ...01098
        DO 102 II=1,NUMNP,50                                             ...01099
        WRITE(6,2000) O,HEAD,TIME,(I,CD(1),CD(2),I=1,NDM),(I,DI(1)       ...01100
       1  ,CD I(2),I=1,NDF)                                              ...01101
        JJ = MIN0(NUMNP,II+49)                                           ...01102
        DO 102 N=II,JJ                                                   ...01103
        IF(FCOMP(X(1,N),BL)) GO TO 101                                   ...01104
        DO 100 I = 1,NDF                                                 ...01105
        UL(I) = F(I,N)*PROP                                              ...01106
        K = IABS(IC)                                                     ...01107
        IF(K.GT.C) LL(I) = B(K)                                         ...01108
    100 WRITE(6,FDIS) N,(X(I,N),I=1,NDM),(UL(I),I=1,NDF)                 ...01109
    101 CONTINUE                                                         ...01110
    102 CONTINUE                                                         ...01111
        RETURN                                                           ...01112
   2000 FORMAT(A1,2CA4//5X,19HNODAL DISPLACEMENTS,5X,4HTIME,E13.5//      ...01113
       1 6X,4HNODE,9(I7,A4,A2))                                          ...01114
        END                                                             ...01115
                                                                         ...01116
        SUBROUTINE PRITE (A,AA,K)                                        ...01117
        IMPLICIT REAL*8 (A-H,O-Z)                                        ...01118
C                                                                        ...01119
C.... OUTPUT ARRAY                                                       ...01120
C                                                                        ...01121
        DIMENSION A(AA),WD(2)                                            ...01122
        WRITE(6,2000) WD                                                 ...01123
        DO 100 J=1,AA,7                                                  ...01124
        K = MIN(AA,J+6)                                                  ...01125
        WRITE(6,2001) (J,A(J),J=1,K)                                     ...01126
        RETURN                                                           ...01127
    100 FORMAT(/1CX,2A8,6H ARRAY /)                                      ...01128
   2000                                                                  ...01129
                                                                         ...01130
```

125

```
2001      FORMAT(1X,7(I5,1PE12.4))
          END

          SUBROUTINE PROFIL (JDIAG,ID,IX,NDF,NEN1,NAD)
          IMPLICIT REAL*8 (A-H,O-Z)
C....
C....     COMPUTE PROFILE OF GLOBAL ARRAYS
C
          COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
          DIMENSION JDIAG(1),ID(NDF,1),IX(NEN1,1)
C....     SET UP THE EQUATION NUMBERS
          NEQ = 0
          DO 50 N = 1,NUMNP
          DO 40 I = 1,NDF
          J=ID(I,N)
          IF(J)30,20,30
20        NEQ = NEQ + 1
          ID(I,N) = NEQ
          JDIAG(NEQ) = 0
          GO TO 40
30        ID(I,N) = 0
40        CONTINUE
50        CONTINUE
C....     COMPUTE COLUMN HEIGHTS
          DO 500 N = 1,NUMEL
          DO 400 I = 1,NEN
          II = IABS(IX(I,N))
          IF(II.EQ.0) GO TO 400
          DO 300 K = 1,NDF
          KK = IC(K,II)
          IF(KK.EQ.0) GO TO 300
          DO 200 J = 1,NEN
          JJ = IABS(IX(J,N))
          IF(JJ.EQ.0) GO TO 200
          DO 100 L = 1,NDF
          LL = IC(L,JJ)
          IF(LL.EQ.0) GO TO 100
          M = MAX0(KK,LL)
          JDIAG(M) = MAX0(JDIAG(M),IABS(KK-LL))
100       CONTINUE
200       CONTINUE
300       CONTINUE
400       CONTINUE
500       CONTINUE
C....     COMPUTE DIAGONAL POINTERS FOR PROFILE
          NAD = 1
          JDIAG(1) = 1
```

126

```
      IF(NEC.EC.1) RETURN                                            :::01179
600   JDIAG(N) = JDIAG(N) + JDIAG(N-1) + 1                           :::01180
      NAC = JDIAG(NEQ)                                               :::01181
      RETURN                                                         :::01182
      END                                                           :::01183
                                                                    :::01184
C.....                                                               :::01185
C     FORM LCAC VECTOR IN CCMPACT FORM                               :::01186
      SUBROUTINE PLOAD(IC,F,B,NN,P)                                  :::01187
      IMPLICIT REAL*8 (A-H,O-Z)                                      :::01188
      DIMENSION ID(1),F(1),B(1)                                      :::01189
      DO 100 N = 1,NN                                                :::01190
      J = IC(N)                                                      :::01191
10C   IF(J.GT.C) B(J) = F(N)*P                                       :::01192
      RETURN                                                         :::01193
      END                                                            :::01194
                                                                    :::01195
C.....                                                               :::01196
C     COMPUTE ELEMENTS  ARRAYS AND ASSEMBLE GLOBAL ARRAYS            :::01197
      SUBROUTINE PFORM(UL,XL,TL,LD,P,S,IE,D,IC,X,IX,F,T,JDIAG,B,A,C,NDF, :::01198
     1NEM,NEN1,NST,ISW,U,UD,AFL,BFL,CFL,DFL,ITP)                     :::01199
      IMPLICIT REAL*8 (A-H,O-Z)                                      :::01200
      LOGICAL AFL,BFL,CFL,DFL                                        :::01201
      COMMCN /CCATA/ O,HEAC(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR       :::01202
      COMMCN /ELCATA/ DM,A,NA,ACT,IEL,NEL                            :::01203
      COMMCN /PRLOD/ PROP                                            :::01204
      DIMENSION XL(NDM,1),LC(NEF,1),P(1),S(NST,1),IE(7,1),D(10,1),ID(NCF, :::01205
     1,1),X(NCN,1),IX(NEN1,1),F(NCF,1),JDIAG(1),B(1),A(1),C(1),UL(NCF,1), :::01206
     2,TL(1),U(1),UC(1),LC(NCF,1)                                    :::01207
C....                                                                :::01208
C     CN ELEMENTS                                                    :::01209
      LCCF = CFL                                                     :::01210
      IEL = 11C                                                      :::01211
      DO 11C N = 1,NUMEL                                             :::01212
      MA = IX(NEN1,N)                                                :::01213
C....                                                                :::01214
C     SET UP LOCAL ARRAYS                                            :::01215
      DO 1C8 I = 1,NEN                                               :::01216
      II = IABS(IX(I,N))                                             :::01217
      IF(II.NE.C) GO TO 1C5                                          :::01218
      TL(I) = C.OC0                                                  :::01219
      DO 1C3 J = 1,NDM                                               :::01220
      XL(J,I) = 0.OC0                                                :::01221
      DO 1C4 J = 1,NDF                                               :::01222
      UL(J,I) = C.CDO                                                :::01223
103   UL(J,I+NEN) = 0.OC0                                            :::01224
                                                                    :::01225
                                                                    :::01226
```

127

```
104    LD(J,I) = 0
       GO TO 108
105    IID = II*NDF - NDF
       NEL = I
       TL(I) = T(II)
       DO 106 J = 1,NDM
       XL(J,I) = X(J,II)
106    CONTINUE
       JJ = IE(J,MA)*NDF
       IF(JJ.LE.0) GO TO 107
       K = IE(IE(JJ,II))*PROP
       UL(J,I+NEN) = UD(J,II)
       UL(K.GT.C) UL(J,I) = L(K)
       IF(DFL) IID = IID + JJ
107    LD(J,I) = K
       LD(IX(I,N).GT.0) GO TO 108
       DO 109 J = 1,NDF
       LD(J,I) = -LD(J,I)
108    CONTINUE
C.....
       FORM ELEMENT ARRAY    MCT = 0
       IF(IE(7,MA).NE.IEL)
       CALL ELMLIB(D(1,MA),UL,XL,IX(1,N),TL,S,P,NCF,NDM,NST,ISW)
C.....
       ADD TO TOTAL ARRAY
       IF(AFL.OR.CFL.OR.CFL) CALL ADDSTF(A,B,C,S,P,JDIAG,LD,NST,NEL*NDF,
      1 AFL,CFL,BFL,CFL,ITP)
110    CONTINUE
       END FILE ITP
       RETURN
       END

       SUBROUTINE PEIGS(A,B,F,X,Y,Z,ID,IX,JDIAG,NDF,NDM,NEN1,DFL)
       IMPLICIT REAL*8 (A-F,O-Z)
C.....
C.....COMPUTE DOMINANT EIGENVALUE BY INVERSE ITERATION
C
       LOGICAL CFL
       COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
       COMMON /ENGYS/ AENGY
       DIMENSION A(1),B(1),F(1),X(1),Y(1),Z(1),ID(1),IX(1),JDIAG(1)
       DATA ITS/ICO/,TOL/1.CD-S/
C.....
       SET START VECTOR FROM DIAGONAL OF MASS MATRIX
       DO 100 I = 1,NEQ
       J = JDIAG(I)
       IF(DFL) J = I
100    Y(I) = B(J)
```

```
      EIGP = 0.0D0
      CALL ACTCCL(A,Z,JDIAG,NEG,.TRUE.,.FALSE.)
  200 IF( II.EQ.1 ) ITS
      CALL PZERC(Z,NEG*IFR)
C.... RAYLEIGH QUOTIENT
      EIG = AENGY/DOT(Y,Z,NEG)
      IF(DABS((EIG-EIGP).LT.TOL*DABS(EIG)) GO TO 300
      CALL NORM(Y,Z,NEQ)
      EIGP = EIG
C.... INVERSE ITERATION
  200 CALL ACTCCL(A,Y,JDIAG,NEG,.FALSE.,.TRUE.)
      WRITE(6,2001) ITS
      RETURN
  300 WRITE(6,2000) O,HEAD,EIG,I
      CALL NORM(Z,Y,NEQ)
      CALL PRTCIS(ID,X,Z,F,NDM,NDF)
      RETURN
 2000 FORMAT(A1,20A4//5X,14HEIGENVALUE =    ,G13.4/5X,14HITERATICNS =    ,
     1   I9/)
 2001 FORMAT(5X,57H**FATAL ERRCR 09** NO CCNVERGENCE IN EIGENVALUES, ITS
     1  = ,I5)
      END

C.....
C
      SUBROUTINE PRTREA(R,NDF)
      IMPLICIT REAL*8(A-H,O-Z)
      PRINT NODAL REACTICNS
C
      DIMENSION R(NDF,1),RSUM(6),ASUM(6)
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      DO 50 K=1,NDF
      RSUM(K) = 0.0D0
      ASUM(K) = 0.0D0
   50 DO 100 N=1,NUMNP,50
      J=MIN0(NUMNP,N+49)
      WRITE(6,2000) O,HEAD,(K,K=1,NDF)
      DO 100 I=N,J,1
      DO 75 K = 1,NDF
      R(K,I) = -R(K,I)
      RSUM(K) = RSUM(K) + R(K,I)
   75 ASUM(K) = ASUM(K) + DABS(R(K,I))
  100 WRITE(6,2001) I,(R(K,I),K=1,NDF)
C.... PRINT STATICS CHECK
      WRITE(6,2002) (RSUM(K),K=1,NDF)
      WRITE(6,2003) (ASUM(K),K=1,NDF)
      RETURN
 2000 FORMAT(A1,2CA4//5X,15HNODAL REACTIONS//6X,4HNODE,
```

```
      1 6((19,4H CCF))
2001  FORMAT(I1C,6E13.4)                                            ::001323
2002  FORMAT(/7X,3HSUM,6E12.4)                                      ::001324
2003  FORMAT(/3X,7HABS.SUM,6E13.4)                                  ::001325
      END                                                           ::001326
                                                                    ::001327
      SUBRCUTINE PSETM(NA,NE,NJ,AFL)                                ::001328
      IMPLICIT REAL*8 (A-H,C-Z)                                     ::001329
C....                                                               ::001330
C.... SET FCINTER FOR ARRAYS                                        ::001331
C                                                                   ::001332
      LOGICAL AFL                                                   ::001333
      NA = NE + NJ                                                  ::001334
C**   NE = NA + NCD((NA-1),IPR)                                     ::001335
      AFL = .FALSE.                                                 ::001336
      CALL SETMEM(NE)                                               ::001337
      RETURN                                                        ::001338
      END                                                           ::001339
                                                                    ::001340
      SUBRCUTINE PZERO(V,NN)                                        ::001341
C....                                                               ::001342
C.... ZERC REAL ARRAY                                              ::001343
C                                                                   ::001344
      DIMENSICN V(NN)                                               ::001345
      DO 1CC N = 1,NN                                               ::001346
100   V(N) = C.0D0                                                  ::001347
      RETURN                                                        ::001348
      END                                                           ::001349
                                                                    ::001350
      SUBRCUTINE UACTCL(A,C,B,JDIAG,NEQ,AFAC,BACK)                  ::001351
      IMPLICIT REAL*8 (A-H,O-Z)                                     ::001352
      LOGICAL AFAC,BACK                                             ::001353
      DIMENSION A(1),B(1),JDIAG(1),C(1)                             ::001354
C....                                                               ::001355
C.... UNSYMMETRIC, ACTIVE CCLUMN PRCFILE EQUATICN SOLVER            ::001356
C....                                                               ::001357
C.... FACTCR A TC UT*D*U, RECUCE B TO Y                             ::001358
      JR = C                                                        ::001359
      DO 300 J = 1,NEQ                                              ::001360
      JD = JDIAG(J)                                                 ::001361
      JH = JD - JR                                                  ::001362
      IF(JH.LE.1) GO TO 300                                         ::001363
      IS = J + 1 - JH                                               ::001364
      IE = J - 1                                                    ::001370
```

```fortran
      IF(.NOT.AFAC) GO TO 250
      K = JR + 1
C.... REDUCE ALL EQUATIONS EXCEPT DIAGONAL
      ID = ID
      DO 200 I = IS,IE
      IR = IC
      IH = JDIAG(I)
      IH = MIN0(IC - IR -1,I - IS)
      IF(IH.EG.C) GO TO 150
      K = K + 1
      A(K) = A(K) - DOT(A(K-IH),C(ID-IH),IH)
150   C(K) = C(K) - DOT(C(K-IH),A(ID-IH),IH)
200   IF(A(ID).NE.C.0D0) C(K) = C(K)/A(ID)
C.... REDUCE DIAGONAL TERM
      A(JD) = A(JD) - DOT(A(JR+1),C(JR+1),JH-1)
C.... FORWARD RECUCE THE R.H.S.
250   IF(BACK) B(J) = B(J) - DOT(C(JR+1),B(IS),JH-1)
300   JR=JC
C.... BACKSUBSTITUTION
      JR=NEQ
      JD = JDIAG(J)
500   IF(A(JD).NE.0.0D0) B(J) = B(J)/A(JD)
      J = J - 1
      IF(J.LE.C) RETURN
      JR = JDIAG(J)
      IF(JC - JR.LE.1) GO TO 700
      IS = JR + 2
      K = J - IS + 1
600   B(I) = E(I) - A(I+K)*C
      JD = JR
      GO TO 500
700   END
```

```
      SUBROUTINE FCDE1 (CT,XA,XC,XM,XN,U,F,JDIAG,ID,M,NE,NDF,AFR,CFR,DFR  01406
     1,EFR,EER)                                                          01407
      IMPLICIT REAL*8 (A-H,O-Z)                                          01408
C                                                                        01409
C.... FIRST ORDER ORDINARY DIFFERENTIAL EQUATION SOLVER                  01410
C                                                                        01411
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR           01412
      LOGICAL FCCMP,EER,AFR,CFR,DFR,EFR                                  01413
      DIMENSION F(1),U(1),JDIAG(1),M(1),XA(1),XC(1),XM(1),XN(1),ID(1)    01414
     1 ,CT(3),WC(2)                                                      01415
      DATA WC /4HINIT,4HLINE,4HQUAD/                                     01416
      NNEQ = NDF*NUMNP                                                   01417
      REWIND 5                                                           01418
      WRITE (5) (M(I),I=1,NE)                                            01419
      END FILE 5                                                         01420
      DO 100 J=1,3                                                       01421
      IF(FCCMP(CT(1),WD(J))) GO TO 110                                   01422
100   CONTINUE                                                           01423
      EER = .TRUE.                                                       01424
      RETURN                                                             01425
110   EER = .FALSE.                                                      01426
      GO TO (1,2,3),J                                                    01427
1     DO 10 I=1,NNEQ                                                     01428
10    N(I) = IC(I)                                                       01429
      M1 = N1 + MOD((M1-1),IPR)                                          01430
      CALL INIT (XA,XC,XM,XN,F,M,U,JDIAG,NDF,NNEQ)                       01431
      GO TO 30                                                           01432
2     CALL PLINE (XA,XC,XM,XN,F,M,U,JDIAG,AFR,CFR,EFR)                   01433
      CALL TEFO (XA,XC,XM,XN,F,M,U,JDIAG,AFR,CFR,EFR)                    01434
      GO TO 30                                                           01435
      M2 = 1 + NEQ*IPR                                                   01436
3     N2 = N2 + MOD((M2-1),IPR)                                          01437
      CALL PCLAC (XA,XC,XM,XN,F,M,M(M2),U,JDIAG,AFR,CFR,DFR,EFR)         01438
      CALL TERM (XA,XC,XM,XN,F,M,M(M2),U,JDIAG,AFR,CFR,DFR,EFR)          01439
30    REWIND 5                                                          01440
      READ (5) (M(I),I=1,NE)                                            01441
      RETURN                                                            01442
      END                                                               01443
C                                                                        01444
      SUBROUTINE INIT (XA,XC,XM,XN,F,ID,UP1,U,JDIAG,NDF,NNEQ)            01445
      IMPLICIT REAL*8 (A-H,O-Z)                                          01446
C                                                                        01447
C.... INPUT INITIAL CONDITIONS                                           01448
C                                                                        01449
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR           01450
      COMMON /TDATA/ TIME,CT,C1,C2,C3,C4,C5                              01451
                                                                         01452
                                                                         01453
```

132

```fortran
      COMMON /CDATA/ DUMAX,DUMIN,CONTR
      DIMENSION UI(3),XA(1),XC(1),XM(1),XN(1),F(1),UP1(1),U(1),ID(1),
     1 JDIAG(1)
      LOGICAL ERR,INIC,INPUT,AFR,CFR,DFR,EFR,CONTR
      DATA UI/4H    ,4HINI,4HT CI,4HSPL /
      NN = JDIAG(NEQ)
      READ(5,1000) C5,C1,C2,DUMAX,DUMIN
      IF(C5.LE.0.0D00) C5 = 2.0D0/3.0D0
      IF(C1.LE.0.0D00) C1 = 1.5D0
      IF(C2.LE.0.0D00) C2 = .8D0
      WRITE(6,1002) C5,C1,C2
      CONTR = .TRUE.
      IF(DUMAX.LE.0.0D0.OR.DUMIN.LE.0.0D0) CONTR = .FALSE.
      IF(CONTR) WRITE (6,1001) DUMAX,DUMIN
      DO 9 I=1,NEQ
    9 UP1(I) = 0.0D0
      INPUT = .TRUE.
      CALL GENVEC(NDF,UP1,UI,.TRUE.,ERR,.FALSE.)
      DO 10 I=1,NNEQ
      K = IABS(ID(I))
      IF(K.GT.0) U(K) = UP1(I)
   10 CONTINUE
      WRITE FILE 10) (U(J),J=1,NEQ)
      REWIND FILE 10
      RETURN
 1000 FORMAT(5F10.0)
 1001 FORMAT(5X,8HDUMAX = ,1PG13.5,8HDUMIN = ,1PG13.5)
 1002 FORMAT(5X,8HTHETA = ,1PG13.5,8HGAMMA = ,1PG13.5,7HBETA = ,1PG13.5)
      END

      SUBROUTINE PLINE (XA,XC,XM,XN,F,UP1,U,JDIAG,AFR,CFR,DFR,EFR)
      IMPLICIT REAL*8 (A-H,O-Z)
C.....
C      TWO POINT SCHEME
C
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      COMMON /TDATA/ TIME,DT,C1,C2,C3,C4,C5
      DIMENSION XA(1),XC(1),XM(1),XN(1),F(1),UP1(1),U(1),JDIAG(1)
      LOGICAL AFR,CFR,DFR,EFR
      NN = JDIAG(NEQ)
      REWIND FILE 10
      WRITE FILE 10) (XA(J),J=1,NN)
      END FILE 10
      DO 11 I=1,NEQ
   11 UP1(I) = DT*F(I)
      CC = 1.0D0/(C5*DT)
```

```
      IF(DFR) CALL PROMUL (XM,U,UP1,JDIAG,NEQ)
      IF(EFR) CALL PRODIA (XN,U,UP1,NEQ)
      CALL FRCMUL (LP1,CC,NEQ)
      IF(DFR) CALL PADDM (XA,U,UP1,JDIAG,NEQ)
      IF(DFR) CALL PADDD (XA,XN,CC,NN)
      CALL ACTCOL (XA,UP1,JDIAG,NEQ,.TRUE.)
      REWIND (10)
      READ (10) (XA(J),J=1,NN)
      RETURN
      END
C
      SUBROUTINE PQUAD (XA,XC,XM,XN,F,UM1,UP1,U,JDIAG,AFR,CFR,EFR)
      IMPLICIT REAL*8 (A-H,O-Z)
C.... THREE POINT SCHEME
C
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      COMMON /TDATA/ TIME,CT,C1,C2,C3,C4,C5
      DIMENSION XA(1),XC(1),XM(1),XN(1),F(1),UM1(1),U(1),UP1(1),JDIAG(1)
      LOGICAL AFR,CFR,DFR,EFR,CONTR
      READ (10) (UM1(J),J=1,NEQ)
      NN = JDIAG(NEQ)
      REWIND 10
      WRITE FILE 10
      END FILE 10
      C3 = C1/C3
      C4 = C1/C2
      DO 10 I=1,NEQ
      UP1(I) = 1.0D0/C2*F(I)
      F(I) = F(I) + ((1.0D0-C1)/C3)*U(I)
     1  + ((1.0D0-C1)/C3)*UM1(I)
   10 CONTINUE
      IF(DFR) CALL PROMUL (XM,F,UP1,JDIAG,NEQ)
      IF(EFR) CALL PRODIA (XN,F,UP1,NEQ)
      DO 11 I=1,NEQ
      F(I) = (2.0D0+(.5D0-C1)/C2)*U(I)
     1  + ((C1-.5D0)/C2-1.0D0)*UM1(I)
   11 CONTINUE
      CALL FRCMUL (XA,F,UP1,JDIAG,NEQ)
      IF(DFR) CALL PADDM (XA,XM,C4,NN)
      IF(EFR) CALL PADDD (XA,XN,JDIAG,C4,NEQ)
      CALL ACTCOL (XA,UP1,JDIAG,NEQ,.TRUE.)
      REWIND (10)
      READ (10) (XA(J),J=1,NN)
      RETURN
```

134

```
      END
      SUBROUTINE TERM (XA,XC,XN,XN,F,UP1,U,JDIAG,AFR,CFR,DFR,EFR)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NLMAT,NEN,NEQ,IPR
      COMMON /TDATA/ TIME,CT,C1,C2,C3,C4,C5
      DIMENSION XA(1),XC(1),XM(1),XN(1),F(1),U(1),UP1(1),JDIAG(1)
      COMMON /UDATA/ DUMAX,DUMIN,CONTR
      LOGICAL AFR,CFR,DFR,EFR,CCNTR
      IF(.NOT.CCNTR) GO TO 3
      UN = 0.0D0
      DO 11 J = 1,NEQ
   10 JJ = JDIAG(J)
   11 UN = UN + (UP1(JJ)-U(JJ))*(UP1(JJ)-U(JJ))
      UN = DSQRT (UN)
      K = 0
      IF(UN.GT.DUMAX) K = 1
      IF(UN.LT.DUMIN) K = 2
      GO TO (1,2,3),K
    1 CT = CT/2.0D0
      CALL PLINE (XA,XC,XN,XN,F,UP1,U,JDIAG,AFR,CFR,DFR,EFR)
    2 CT = CT*2.0D0
      CALL PLINE (XA,XC,XN,XN,F,UP1,U,JDIAG,AFR,CFR,DFR,EFR)
    3 REWIND 10
      WRITE (10) (U(J),J=1,NEQ)
      END FILE 10
      DO 14 I=1,NEQ
   14 U(I) = UP1(I)
 1000 WRITE(6,1000) DT
      FORMAT(10X,4HDT =,1PE13.6)
      RETURN
      END
```

```
      SUBROUTINE ELMT02(D,UL,XL,IX,TL,S,P,NDF,NEM,NST,ISW)                01583
      IMPLICIT REAL*8 (A-H,O-Z)                                          01584
C.... TWO DIMENSIONAL HEAT TRANSFER ELEMENT                              01585
C                                                                        01586
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR           01587
      COMMON/ELDATA/ DM,N,MA,MCT,IEL,NEL                                 01588
      DIMENSION D(1),UL(1),XL(NDM,1),IX(NEM,1),IX(1),TL(1),S(NST,1),P(1),01589
     1SHP(3,9),WLAB(4)                                                   01590
      DATA WLAB/4H PLA,4H-NE  ,4HAXIS,4HYM  /                            01591
C                                                                        01592
C.... TRANSFER TO CORRECT PROCESSOR                                      01593
      GO TO (1,2,3,2,5,3),ISW                                            01594
C                                                                        01595
C.... INPUT MATERIALS PROPERTIES                                         01596
    1 NLBC = 0                                                           01597
      READ(5,1000) D(1),D(2),D(3),D(4),D(5),NGP,KAT,NLBC,INQL            01598
      IF(NGP.LE.0.OR.NGP.GT.6) NGP = 4                                   01599
      IF(KAT.NE.2) KAT = 1                                               01600
      WRITE(6,2000) D(1),D(2),D(3),D(4),D(5),NGP,NLBC                    01601
      L(3) = C(KAT)*C(4)                                                 01602
      IF(KAT.EQ.2) WRITE(6,2001) WLAB(3),WLAB(4)                         01603
      IF(KAT.EQ.1) WRITE(6,2001) WLAB(1),WLAB(2)                         01604
      IKX = 0                                                            01605
      IKY = 0                                                            01606
      IRCC = 0                                                           01607
      IQ = 0                                                             01608
C                                                                        01609
      IF(INQL.EQ.0) GO TO 11                                             01610
      READ(5,1001) IKX,IKY,IRCC,IQ                                       01611
      IF(IKX.NE.0) CALL PKX (T,D(1),1)                                   01612
      IF(IKY.NE.0) CALL PKY (T,D(2),1)                                   01613
      IF(IRCC.NE.0) CALL PRCC (T,D(3),1)                                 01614
      IF(IQ.NE.0) CALL PQ (T,D(5),1)                                     01615
   11 CONTINUE                                                           01616
      IF(NLBC.NE.0) CALL ECCND2 (UL,XL,IX,S,P,NDF,NDM,NST,NLBC,NGP,1)    01617
      RETURN                                                             01618
C                                                                        01619
C.... INSERT CHECK OF MESH IF DESIRED                                    01620
    2 RETURN                                                             01621
C                                                                        01622
C.... COMPUTE CONDUCTIVITY (STIFFNESS) MATRIX                            01623
    3 DO 102 II=1,NGP                                                    01624
      CALL FGAUSS(SG,WS,NGP,II)                                          01625
      DO 102 JJ=1,NGP                                                    01626
      CALL FGAUSS(TG,WT,NGP,JJ)                                          01627
      WG = WS*WT                                                         01628
      CALL SHAPE(SG,TG,XL,SHP,XSJ,NDM,NEL,IX,.FALSE.)                    01629
      IF(IKX.EQ.0.AND.IKY.EQ.0.AND.IQ.EQ.0) GO TO 113                    01630
```

```
              T = C*CCO
          DO 111 KK=1,NEL
              T = T + SHP(3,KK)*UL(KK)
          IF(IKX.NE.0) CALL PKX(T,C(1),2)
          IF(IKY.NE.0) CALL PKY(T,C(2),2)
113       IF(IH.NE.0) CALL PQ(T,D(5),2)
100       CONTINUE
          IF(KAT.NE.2) GO TO 101
          RR = RR + SHP(3,I)*XL(1,I)
          CO 100 J = 1,NEL
          XSJ = XSJ*RR
          CV = XSJ*WG
          DO 102 I = 1,NEL
          SHJ = SHP(3,J)*DV
          A1 = D(1)*SHP(1,J)*DV
          A2 = D(2)*SHP(2,J)*DV
          P(J) = F(J)*NEL
102       S(I,J) = S(I,J) + A1*SHP(1,I) + A2*SHP(2,I)
          IF(NLBC.NE.0) CALL BCCNC2 (UL,XL,IX,S,P,NCF,NDM,NST,NLBC,NGP,2)
          DO 106 I = 1,NEL
          CO 106 J = 1,NEL
106       P(I) = P(I) - S(I,J) * UL(J)
          IF(ISW.EQ.3) RETURN
          DO 116 I = 1,NEL
          CO 116 J = 1,NEL
116       S(I,J) = 0.0C0
C.....    COMPUTE HEAT CAPACITY (MASS) MATRIX
5         DO 105 II = 1,NGP
          CALL PGAUSS(SG,WS,NGF,II)
          CO 105 JJ=1,NGP
          CALL PGAUSS(TG,WT,NGF,JJ)
          WG = WS*WT*APE(SG,TG,XL,SHP,XSJ,NDM,NEL,IX,.FALSE.)
          CALL STAPE(SG,TG,XL,SHP,XSJ,NDM,NEL,IX,.FALSE.)
          IF(IRC.EQ.0) GO TC 114
          T = C*CCO
          CO 112 KK=1,NEL
          T = T + SHP(3,KK)*UL(KK)
114       CALL PRCC(T,D(3),2)
          CONTINUE
          IF(KAT.NE.2) GO TC 1C4
          RR = RR + SHP(3,I)*XL(1,I)
          CO 1C3 I = 1,NEL
          XSJ = XSJ*RR
104       CV = XSJ*WG
```

137

```
      DO 105 J=1,NEL                                              01679
      SHJ = C(2)*SHP(3,J)*CV                                      01680
      IF(ISW.EQ.5) P(J) = P(J) + SHJ                              01681
      DO 105 I=1,NEL                                              01682
  105 S(I,J) = S(I,J) + SHJ*SHF(3,I)                              01683
      IF(ISW.EQ.5) RETURN                                         01684
      DO 601 J=1,NEL                                              01685
      DO 601 I=1,NEL                                              01686
  601 P(I) = P(I) - S(I,J)*TL(J)                                  01687
      RETURN                                                      01688
C.....                                                            01689
 1000 FORMAT(5F10.0,4I5)                                          01690
 1001 FORMAT(4I5)                                                 01691
 2000 FORMAT(5X,26HHEAT CONDUCTION ELEMENT    //, 7X,             01692
     1/7X,16HCONDUCTIVITY ,5X,4HKXX = ,1PG15.7, 5X,4HKY =,1PG15.7,01693
     2/7X,13HSPECIFIC HEAT ,1PG15.7,/7X,                          01694
     3/19H+DENSITY ,1PG15.7,/7X,                                  01695
     4/19H+HEAT GENER/UNIT VOL ,1PG15.7,/7X,                      01696
     5 I3,15H GAUSS PTS/CIR ,/7X,                                 01697
      I3,34H LINES WITH SPECIFIED BOUNDARY CONDITIONS)            01698
 2001 FORMAT(6X,2A4,9H ANALYSIS )                                 01699
      END                                                         01700
C.....                                                            01701
      SUBROUTINE SHAPE(SS,TT,X,SHP,XSJ,NDM,NEL,IX,FLG)            01702
      IMPLICIT REAL*8 (A-H,O-Z)                                   01703
C.....                                                            01704
C.... SHAPE FUNCTION ROUTINE FOR TWO DIMENSIONAL ELEMENTS         01705
C.....                                                            01706
      LOGICAL FLG                                                 01707
      COMMON /JAC2L/ XS                                           01708
      DIMENSION SHP(3,1),X(NDM,1),S(4),T(4),XS(2,2),SX(2,2),IX(1) 01709
      DATA S/-0.5D0,0.5D0,0.5D0,-0.5D0/,T/-0.5D0,-0.5D0,0.5D0,0.5D0/ 01710
C.... FORM 4-NODE QUADRILATERAL SHAPE FUNCTIONS                   01711
  100 DO 100 I = 1,4                                              01712
      SHP(3,I) = (0.5D0+S(I)*SS)*(0.5D0+T(I)*TT)                  01713
      SHP(1,I) = S(I)*(0.5D0+T(I)*TT)                             01714
      SHP(2,I) = T(I)*(0.5D0+S(I)*SS)                             01715
      IF(NEL.GE.4) GO TO 120                                      01716
C.... FORM TRIANGLE BY ADDING THIRD AND FOURTH TOGETHER           01717
  110 SHP(1,3) = SHP(1,3)+SHP(I,4)                                01718
C.... ADD QUADRATIC TERMS IF NECESSARY                            01719
      IF(NEL.GT.4) CALL SHAP2(SS,TT,SHP,IX,NEL)                   01720
C.... CONSTRUCT JACOBIAN AND ITS INVERSE                          01721
      DO 130 I = 1,NDM                                            01722
      DO 130 J = 1,2                                              01723
```

138

```fortran
      XS(I,J) = C.0D0
130   DO 130 K = 1,NEL
      XS(I,J)=XS(I,J) + X(J,K)*SHP(I,K)
      IF(FLG) RETURN
      SXX(1,1)= XS(2,2)/XSJ
      SXX(1,2)=-XS(1,2)/XSJ
      SXX(2,1)=-XS(2,1)/XSJ
      SXX(2,2)= XS(1,1)/XSJ
C.... FORM GLOBAL DERIVATIVES
      DO 140 I = 1,NEL
      TP      = SHP(1,I)*SX(1,1)+SHP(2,I)*SX(2,1)
      SHP(2,I)= SHP(1,I)*SX(1,1)+SHP(2,I)*SX(2,2)
      SHP(1,I)= TP
140   RETURN
      END

      SUBROUTINE SHAP2(S,T,SHP,IX,NEL)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C.... ADD QUADRATIC FUNCTIONS AS NECESSARY
C
      DIMENSION IX(1),SHP(3,1)
      S2=(1.0D0-S*S)/2.0D0
      T2=(1.0D0-T*T)/2.0D0
100   CONTINUE
C.... ADD QUADRATIC (SERENDIPITY)
      IF(IX(5).EQ.0) GO TO 101
      SHP(1,5)=S2*(1.0D0-T)
      SHP(2,5)= -S2*(1.0D0-T)
      IF(NEL.LT.6) GO TO 107
101   IF(IX(6).EQ.0) GO TO 102
      SHP(1,6)=T2*(1.0D0+S)
      SHP(2,6)= S2*(1.0D0+S)
      IF(NEL.LT.7) GO TO 107
102   IF(IX(7).EQ.0) GO TO 103
      SHP(1,7)=S2*(1.0D0+T)
      SHP(2,7)= -S2*(1.0D0+T)
      IF(NEL.LT.8) GO TO 107
103   IF(IX(8).EQ.0) GO TO 104
      SHP(1,8)=T2*(1.0D0+T)
      SHP(2,8)= -T*(1.0D0-S)
```

139

```
C.....
104       SHP(3,8) = T2*(1.0D0-S)
C.....    INTERIOR NODE (LAGRANGIAN)
          IF(KEL.LT.9) GO TO 107
          IF(IX(9).EQ.0) GO TO 107
          SHP(1,9) = -S*T2
          SHP(2,9) = -T*S2
C.....    CORRECT EDGE NODES FOR INTERIOR NODE (LAGRANGIAN)
          DO 105 I = 1,2
          SHP(I,9) = 4.0D0*S2*T2
105       SHP(J,I) = SHP(J,I) - 0.25D0*SHP(J,9)
          DO 106 J = 1,2
106       SHP(J,I) = SHP(J,I) - 0.25D0*SHP(J,9)
C.....    CORRECT CORNER NODES FOR PRESENCE OF MIDSIDE NODES
107       IF(IX(I).NE.0) SHP(J,I) = SHP(J,I) - 0.5D0*SHP(J,9)
          K = 8
          DO 108 I = 1,4
          L = I + 4
          DO 108 J = 1,3
108       SHP(J,I) = SHP(J,I) -0.5D0*(SHP(J,K)+SHP(J,L))
109       K = L
          RETURN
          END

          SUBROUTINE BCOND2 (UL,XL,IX,S,P,NDF,NEN,NST,NLBC,NCP,IFLAG)
          IMPLICIT REAL*8 (A-H,O-Z)
          COMMON /ELDATA/ DM,N,NCT,NUMNP,NUMEL,NUMAT,NEN,NEQ,IPR
          COMMON /CDATA/ O,HEAD(20)
          DIMENSION CL(1),XL(NCM,1),IX(1),S(NST,1),P(1),SHP(3,9)
          DIMENSION ICN,KEL(50),KBCOND(50),KLINE(50),PROPE(50,2)
          GO TO (1,2),IFLAG
C.....
1         INPUT LINE BOUNDARY CONDITIONS
          WRITE(6,2000) O,HEAD
          KBMAX = 0
          DO 10 I=1,NLBC
          READ (5,1000) KEL(I),KBCOND(I),KLINE(I),PROPE(I,1),PROPB(I,2)
          KBMAX = MAXO(KBMAX,KBCOND(I))
10        WRITE(6,2001) KEL(I),KBCOND(I),KLINE(I),PROPE(I,1),PROPB(I,2)
          IF(KBMAX.GE.4) CALL CONV(T,COEFF,1)
          RETURN
C.....
C.....    ADD LINE BOUNDARY CONDITIONS CONTRIBUTIONS
2         TO CONDUCTIVITY MATRIX AND LOAD VECTOR
          DO 30 LL=1,NLBC
          IF(KEL(LL).NE.N) GO TO 30
          KBT = KBCOND(LL)
          KBL = KLINE(LL)
```

140

```
      CONST = PROPB(LL,1)
      TEMF = FFCFE(LL,2)
      KBLABS = IABS(KBL)
      C2 = CFLCAT(KBL/KBLABS)
      DO 30 III=1,NGP,II
      CALL PGAUSS (C1,W1,NGP,II)
      WG = W1
C.... SELECT CORRECT INTEGRATICN POINTS
      GO TC (21,22),KBLABS
21    SG = C1
      TG = C1
      GO TC 24
22    SG = C1
      TG = C1
24    GO TC 24
      CALL SHAPE (SG,TG,XL,SHP,DUM,NDM,NEL,IX,.TRUE.)
      CALL JACE2 (XSJ,KELAES)
      DS = WG*XSJ
      COEFF = CCNST
      GO TC (25,26,27,28),KET
C.... FLUX BCLNCARY CONDITION
25    DO 251 L=1,NEL
251   P(L) = F(L) - COEFF*SHP(3,L)*DS
      GO TC 30
C.... RACIATICN EOUNDARY CCNDITION
27    T = 0.0E0
      DO 271 I=1,NEL
271   T = T + SHP(3,I)*UL(I)
      COEFF = CCEFF*(T*T + TEMF*TEMP)*(T + TEMF)
      GO TC 26
C.... CONVECTICN BCUNDARY CCNDITION (TEMP. CEPEDENT)
28    T = C.CEO
      DO 281 I=1,NEL
281   T = T + SHP(3,I)*UL(I)
      CALL CCNV(T,COEFF,2)
C.... CONVECTICN BOUNDARY CCNDITION
26    DO 261 J=1,NEL
      A = SHF(3,J)*DS
      P(J) = F(J) + COEFF*TEMP*A
      DO 261 I = 1,NEL
261   S(I,J) = S(I,J) + CCEFF*SHP(3,I)*A
30    CONTINLE
      RETURN
```

```
1000  FORMAT (3I5,2F10.0)
2000  FORMAT (A1,20A4,//5X,9HLINE B.C.//
     1 5X,52H  ELEM. B.C.     LINE PROPERTY VALUE       TEMPERATURE)
2001  FORMAT (5X,3(I5,2X),2(1PE15.7,2X))
      END

      SUBROUTINE JACBB2 (XS,J,KL)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C.... COMPUTE LINE JACOBIAN DETERMINANT
C
      COMMON /JAC2L/ XS
      DIMENSION XS(2,2)
      GO TO (1,2),KL
    1 D1 = XS(1,2)
      D2 = XS(2,2)
      GO TO 10
    2 D1 = XS(1,1)
      D2 = XS(1,2)
      GO TO 10
   10 XSJ = DSQRT(C1*D1 + C2*D2 )
      RETURN
      END

      SUBROUTINE PKX (T,C,IFLAG)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C.... CALCULATE CONDUCTIVITY (X DIRECTION)
C
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      DIMENSION TEMP(50),CCEF(50)
      IF(IFLAG.EC.1) WRITE (6,2000) C,HEAD
      CALL TABLE (TEMP,CCEF,T,C,NDATA,IFLAG)
      RETURN
2000  FORMAT(A1,20A4,//5X,26HCONDUCTIVITY DATA (X DIR.)//,
     1 5X,30H       TEMPERATURE           KX )
      END

      SUBROUTINE PKY (T,C,IFLAG)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C.... CALCULATE CONDUCTIVITY (Y DIRECTION)
C
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      DIMENSION TEMP(50),CCEF(50)
```

01871
01872
01873
01874
01875
01876
01877
01878
01879
01880
01881
01882
01883
01884
01885
01886
01887
01888
01889
01890
01891
01892
01893
01894
01895
01896
01897
01898
01899
01900
01901
01902
01903
01904
01905
01906
01907
01908
01909
01910
01911
01912
01913
01914
01915
01916
01917
01918

142

```
         IF(IFLAG.EQ.1) WRITE (6,2000) Q,HEAD
         CALL TABLE (TEMP,COEF,T,C,NDATA,IFLAG)
         RETURN
2000     FORMAT(A1,20A4,//5X,26HCONDUCTIVITY DATA (Y DIR.)//,
        1 5X,3H    TEMPERATURE              KY )
         END

C.....   CALCULATE HEAT CAPACITY
C
         SUBROUTINE PROC (T,C,IFLAG)
         IMPLICIT REAL*8 (A-H,O-Z)
         COMMON /CDATA/ Q,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
         DIMENSION TEMP(50),CCEF(50)
         IF(IFLAG.EQ.1) WRITE (6,2000) Q,HEAD
         CALL TABLE (TEMP,COEF,T,C,NDATA,IFLAG)
         RETURN
2000     FORMAT(A1,20A4,//5X,18HHEAT CAPACITY DATA//,
        1 5X,3H    TEMPERATURE              RO*C )
         END

C.....   CALCULATE INTERNAL HEAT GENERATION
C
         SUBROUTINE PC (T,C,IFLAG)
         IMPLICIT REAL*8 (A-H,C-Z)
         COMMON /CCATA/ Q,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
         DIMENSION TEMP(50),CCEF(50)
         IF(IFLAG.EQ.1) WRITE (6,2000) Q,HEAD
         CALL TABLE (TEMP,COEF,T,C,NDATA,IFLAG)
         RETURN
2000     FORMAT(A1,20A4,//5X,2HINT.HEAT GENER. DATA//,
        1 5X,3H    TEMPERATURE              Q )
         END

C.....   CALCULATE HEAT TRANSFER COEFFICIENT
C
         SUBROUTINE CONV (T,C,IFLAG)
         IMPLICIT REAL*8 (A-H,C-Z)
         COMMON /CCATA/ Q,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
         DIMENSION TEMP(50),CCEF(50)
         IF(IFLAG.EQ.1) WRITE (6,2000) Q,HEAD
         CALL TABLE (TEMP,COEF,T,C,NDATA,IFLAG)
         RETURN
```

```fortran
2000  FORMAT(A1,20A4;//5X,24HHEAT TRANSFER COEF. CATA//,
     1 5X,30H TEMPERATURE      COEFFICIENT)
      END

      SUBROUTINE TABLE (XX,YY,X,Y,N,K)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C.....CALCULATE Y =TABLE(X) BY LINEAR INTERPOLATION
      DIMENSION XX(1),YY(1)
      GO TO (1,2),KK
    1 READ (5,1000) N
      DO 11 I=1,N
      READ (5,1001) XX(I),YY(I)
      WRITE(6,2001) XX(I),YY(I)
   11 RETURN
    2 IF(X.LT.XX(N).AND.X.GT.XX(1)) GO TO 21
      IF(X.GE.XX(N)) Y = YY(N)
      IF(X.LE.XX(1)) Y = YY(1)
      RETURN
   21 DO 22 I=1,N
      IF(X.LE.XX(I)) GO TO 23
   22 CONTINUE
   23 X1 = XX(I)   - X
      X2 = XX(I-1) - X
      Y1 = YY(I)
      Y2 = YY(I-1)
      Y = (X1*Y2 - X2*Y1)/(X1 - X2)
      RETURN
 1000 FORMAT(I5)
 1001 FORMAT(2F10.0)
 2001 FORMAT(5X,2(1PG15.7,2X))
      END
```

01967
01968
01969
01970
01971
01972
01973
01974
01975
01976
01977
01978
01979
01980
01981
01982
01983
01984
01985
01986
01987
01988
01989
01990
01991
01992
01993
01994
01995
01996
01997
01998
01999
02000

144

```
      SUBROUTINE ELMT03 (D,UL,XL,IX,TL,S,P,NDF,NDM,NST,ISW)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C..... THREE DIMENSIONAL HEAT TRANFER ELEMENT
C
C          MATERIAL PROPERTIES
C          D(1) CONDUCTIVITY IN X DIRECTION
C          D(2) CONDUCTIVITY IN Y DIRECTION
C          D(3) CONDUCTIVITY IN Z DIRECTION
C          D(4) SPECIFIC HEAT
C          D(5) MASS DENSITY
C          D(6) HEAT GENERATION PER UNIT VOLUME
C
C          NGP      NUMBER OF INTEGRATION POINTS
C          WG       INTEGRATION WEIGHT
C          KAT     =2 FOR AXISYMMETRY
C                  =1 FOR PLANE GEOMETRY
C          NSBC     NUMBER OF SURFACES WITH SPECIFIED
C                   BOUNDARY CONDITIONS
C
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      COMMON /ELDATA/ DM,N,MA,MCT,IEL,NEL
      DIMENSION D(N),UL(1),XL(NDM,1),IX(NEN,1),TL(1),S(NST,1),P(1),
     1 SHP(4,21),WLAB(4)
      DATA WLAB /4HPLA,4HNE ,4HAXIS,4HYM /
C
C..... TRANFER TO CORRECT PROCESSOR
      GO TO (1,2,3,2,5,3), ISW
C
C..... INPUT MATERIAL PROPERTIES
1     NSBC = 0
      READ(5,1000) D(1),D(2),D(3),D(4),D(5),D(6),NGP,KAT,NSBC,INOL
      IF(NGP.LE.0.OR.D(1).GT.6) NGP = 4
      WRITE(6,2000) D(1),C(2),D(3),D(4),D(5),D(6),NGP,NSBC
      D(4) = D(4)*D(5)
      IF(KAT.EQ.2) KAT = 1
      IF (KAT.EQ.-1) WRITE(6,2001) WLAB(1),WLAB(2)
      IF (KAT.EQ.-2) WRITE(6,2001) WLAB(3),WLAB(4)
      IF(INOL.EQ.0) GO TO 11
      READ(5,1001) IKX,IKY,IKZ,IRDC,IQ
      IF(IKX.NE.0) CALL PKX (T,D(1),1)
      IF(IKY.NE.0) CALL PKY (T,D(2),1)
      IF(IKZ.NE.0) CALL PKZ (T,D(3),1)
      IF(IRDC.NE.0) CALL PRDC (T,D(4),1)
      IF(IQ.NE.0) CALL PG (T,D(6),1)
11    CONTINUE
      IF(NSBC.NE.0) CALL BCOND3 (UL,XL,S,P,NDF,NDM,NST,NSBC,NGP,1)
      RETURN
C
```

::002001
::002002
::002003
::002004
::002005
::002006
::002007
::002008
::002009
::002010
::002011
::002012
::002013
::002014
::002015
::002016
::002017
::002018
::002019
::002020
::002021
::002022
::002023
::002024
::002025
::002026
::002027
::002028
::002029
::002030
::002031
::002032
::002033
::002034
::002035
::002036
::002037
::002038
::002039
::002040
::002041
::002042
::002043
::002044
::002045
::002046
::002047
::002048

```
C.....  INSERT CHECK OF MESH IF DESIRED
       RETURN
C.....
C      COMPUTE CONDUCTIVITY (STIFFNESS) MATRIX
3      DO 1C2 II=1,NGP
       CALL FGALS(RG,WR,NGF,II)
       DO 1C2 JJ=1,NGP
       CALL FGALS(SG,WS,NGF,JJ)
       DO 1C2 KK=1,NGP
       CALL FGALS(TG,WT,NGF,KK)
       WG=WR*WS*WT
       CALL SHAP3D(RG,SG,TG,XL,SHP,XSJ,NDM,NEL,IX,.FALSE.)
       IF(IKX.EQ.0.AND.IKY.EQ.0.AND.IKZ.EQ.0.AND.IQ.EQ.0) GO TO 112
       DO 111 L = 1,NEL
       T = T + SHP(4,L)*UL(L)
111    CONTINUE
       IF(IKX.NE.0) CALL FKX(T,C(1),2)
       IF(IKY.NE.0) CALL FKY(T,C(2),2)
       IF(IKZ.NE.0) CALL FKZ(T,C(3),2)
       IF(IQ.NE.C)  CALL FQ(T,C(6),2)
112    CONTINUE
       IF(KAT.NE.2) GO TO 101
       RR = 0.0
       DO 100 I=1,NEL
       RR = RR + SHP(4,I)*XL(1,I)
100    XSJ = XSJ*RR
101    DV = XSJ*WG
       DV1 = C(1)*DV
       DV2 = C(2)*DV
       DV3 = C(3)*DV
       DO 102 J=1,NEL
       CB11=DV1*SHP(1,J)
       CB21=DV2*SHP(2,J)
       CB31=DV3*SHP(3,J)
       P(J) = P(J) + SHP(4,J)*C(6)*DV
       DO 102 I = 1,NEL
       S(I,J) = S(I,J) + SHP(1,I)*DB11 + SHP(2,I)*DB21 + SHP(3,I)*DB31
102    IF(NSBC.NE.0) CALL BCCND3 (UL,XL,IX,S,F,NCF,NDM,NST,NSBC,NGP,2)
       DO 103 I=1,NEL
103    P(I) = P(I) - S(I,J)*UL(J)
       RETURN
C.....  COMPUTE HEAT CAPACITY (MASS) MATRIX
5      DO 5C2 II=1,NGP
       CALL FGALS(RG,WR,NGF,II)
       DO 5C2 JJ=1,NGP
       CALL FGALS(SG,WS,NGF,JJ)
```

```
      DO 502 KK=1,NGP
      CALL FGAUSS(TG,WT,NGF,KK)
      WG=RR*WT*WT
      CALL STAFEC(RG,SG,TG,XL,SHP,XSJ,NDM,NEL,IX,.FALSE.)
      IF(IRCC.EQ.0) GO TC 512
      T = C.CC
      DO 511 L=1,NEL
511   T = T + SHF(4,L)*UL(L)
      CALL PROC(T,D(4),2)
512   CONTINUE
      IF(KAT.NE.2) GO TO 5C1
      RR=0.0EC
      DO 500 I=1,NEL
500   RR = RR + SHP(4,I)*XL(1,I)
      XSJ = XSJ*RR
      CV = XSJ*WG
      DO 502 J=1,NEL
      SHJ = C(4)*SHP(4,J)*CV
      P(J) = P(J) + SHJ
      DO 502 I=1,NEL
502   S(I,J) = S(I,J) + SHJ*SHF(4,I)
      RETURN
1000  FORMAT(6F10.0,4I5)
1001  FORMAT(5I5)
2000  FORMAT(EX,23HEAT CONDUCTION ELEMENT // 7X,       =,1PG15.7,5X,4HKY =,1PG15.7,5X,
     1    12HCCNCLCTIVITY,5X,4HKX =,1PG15.7,/7X,1SHSPECIFIC HEAT ,1PG15.7,/7X,
     2    4HKZ =,1PG15.7,/7X,19HSPECIFIC HEAT
     3    19HDENSITY
     4    19HHEAT GENER/UNIT VCL,1PG15.7,/7X,I3,15H GAUSS PTS/CIR,/
     5    7X,I3,45H SURFACES WITH SPECIFIED BOUNDARY CONDITIONS)
2001  FORMAT(6X,2A4,9H ANALYSIS)
      END

      SUBROUTINE SHAP3D (RR,SS,TT,X,SHP,XSJ,NCM,NEL,IX,FLG)
      IMPLICIT REAL*8 (A-H,C-Z)
C.....SHAPE FUNCTION ROUTINE FOR THREE DIMENSIONAL ELEMENTS
C.....4-21 NCCES
C
C     RR,SS,TT    NATURAL COORDINATES WHERE INTERPOLATICN
C                 FUNCTIONS ARE EVALUATED
C     X           X CERIVATIVE OF SHAPE FUNCTIONS
C     Y           Y CERIVATIVE OF SHAPE FUNCTIONS
C     Z           Z CERIVATIVE OF SHAPE FUNCTIONS
C     SHP(1,I)    SHAPE FUNCTIONS
C     SHP(2,I)    SHAPE FUNCTIONS
C     SHP(3,I)    SHAPE FUNCTIONS
C     SHP(4,I)    SHAPE FUNCTIONS
C     XSJ         JACOBIAN ARRAY
C                 JACOBIAN DETERMINANT
```

```
C        X(1,I)     X NODAL COORDINATES
C        X(2,I)     Y NODAL COORDINATES
C        X(3,I)     Z NODAL COORDINATES
C
      LOGICAL FLG
      COMMON /JAC3S/ XS
      DIMENSION SHP(4,1),X(NDM,1),R(8),S(8),T(8),XS(3,3),SX(3,3),IX(1)
      DATA R/0.5D0,-0.5D0,-0.5D0,0.5D0,0.5D0,-0.5D0,-0.5D0,0.5D0/,
     1     S/0.5D0,0.5D0,-0.5D0,-0.5D0,0.5D0,0.5D0,-0.5D0,-0.5D0/,
     2     T/0.5D0,0.5D0,0.5D0,0.5D0,-0.5D0,-0.5D0,-0.5D0,-0.5D0/
C.... COMPUTE 8-NODE RECTANGULAR PRISM SHAPE FUNCTIONS AND DERIVATIVES
C.... IN NATURAL COORDINATES
      DO 100 I = 1,8
      SHP(4,I) = (0.5D0+R(I)*RR)*(0.5D0+S(I)*SS)*(0.5D0+T(I)*TT)
      SHP(1,I) = R(I)*(0.5D0+S(I)*SS)*(0.5D0+T(I)*TT)
      SHP(2,I) = (0.5D0+R(I)*RR)*S(I)*(0.5D0+T(I)*TT)
      SHP(3,I) = (0.5D0+R(I)*RR)*(0.5D0+S(I)*SS)*T(I)
  100 IF(NEL.GE.8) GO TO 200
C.... DEGENERATE FORMS
C
C.... ADD QUADRATIC TERMS IF NECESSARY
  200 IF(NEL.GT.8) CALL SHAP32(RR,SS,TT,SHP,IX,NEL)
C.... CONSTRUCT JACOBIAN AND ITS INVERSE
  201 DO 300 I = 1,NDM
      DO 300 J = 1,3
      XS(I,J) = 0.0D0
      DO 300 K = 1,NEL
  300 XS(I,J) = XS(I,J) + X(J,K)*SHP(I,K)
      IF(FLG) RETURN
      XSJ = XS(1,1)*XS(2,2)*XS(3,3) + XS(1,2)*XS(2,3)*XS(3,1)
     1    + XS(1,3)*XS(2,1)*XS(3,2) - XS(1,3)*XS(2,2)*XS(3,1)
     2    - XS(1,1)*XS(2,3)*XS(3,2) - XS(1,2)*XS(2,1)*XS(3,3)
      IF(XSJ.GT.1.0D-13) GO TO 400
      WRITE(6,1000)
      STOP
  400 SX(1,1) = (XS(2,2)*XS(3,3)-XS(2,3)*XS(3,2))/XSJ
      SX(1,2) = (XS(1,3)*XS(3,2)-XS(1,2)*XS(3,3))/XSJ
      SX(1,3) = (XS(1,2)*XS(2,3)-XS(1,3)*XS(2,2))/XSJ
      SX(2,1) = (XS(2,3)*XS(3,1)-XS(2,1)*XS(3,3))/XSJ
      SX(2,2) = (XS(1,1)*XS(3,3)-XS(1,3)*XS(3,1))/XSJ
      SX(2,3) = (XS(1,3)*XS(2,1)-XS(1,1)*XS(2,3))/XSJ
```

148

```fortran
      SX(2,3) = -(XS(1,1)*XS(2,3)-XS(2,1)*XS(1,3))/XSJ
      SX(3,3) =  (XS(1,1)*XS(2,2)-XS(2,1)*XS(1,2))/XSJ
C.... FORM GLOBAL DERIVATIVES
      DO 500 I=1,NEL
      TP1 = SHP(1,I)*SX(1,1) + SHP(2,I)*SX(1,2) + SHP(3,I)*SX(1,3)
      TP2 = SHP(1,I)*SX(2,1) + SHP(2,I)*SX(2,2) + SHP(3,I)*SX(2,3)
      SHP(3,I)= SHP(1,I)*SX(3,1) + SHP(2,I)*SX(3,2) + SHP(3,I)*SX(3,3)
      SHP(1,I) = TP1
500   SHP(2,I) = TP2
      RETURN
1000  FORMAT(5X,'**FATAL ERROR   ** NEGATIVE OR ZERO JACOBIAN DETERMINAN
     1T.')
      END

      SUBROUTINE SHAP32(R,S,T,SHP,IX,NEL)
      IMPLICIT REAL*8 (A-H,O-Z)
C.... ADD QUADRATIC FUNCTIONS AS NECESSARY
      DIMENSION IX(1),SHF(4,1)
      DATA ONE/1.0D0/,PFIVE/0.5D0/
      R2 = (ONE-R*R)/4.0D0
      S2 = (ONE-S*S)/4.0D0
      T2 = (ONE-T*T)/4.0D0
      DO 100 J=1,4
100   SHF(J,1) = 0.0D0
C.... MIDSIDE NODES (SERENDIPITY)
      IF(IX(1).LT.10) GO TO 101
      SHP(1,1) = -R*PFIVE*(ONE+S)*(ONE+T)
      SHP(2,1) =   R2*(ONE+S)*(ONE+T)
      SHP(3,1) =   R2*(ONE+S)*(ONE+T)
      SHP(4,1) =   R2*(ONE+S)*(ONE+T)
101   IF(NEL.LT.10) GO TO 102
      IF(IX(1).LT.100) GO TO 200
      SHP(1,1) = -S2*(ONE+T)
      SHP(2,1) = -S2*PFIVE*(ONE-R)*(ONE+T)
      SHP(3,1) =   S2*(ONE-R)*(ONE+T)
      SHP(4,1) =   S2*(ONE-R)*(ONE+T)
102   IF(NEL.LT.11) GO TO 103
      IF(IX(1).LT.11) GO TO 200
      SHP(1,1) = -R*PFIVE*(ONE-S)*(ONE+T)
      SHP(2,1) = -R2*(ONE+T)
      SHP(3,1) =   R2*(ONE-S)*(ONE+T)
```

```
      SHP(4,11)=R2*(ONE-S)*(ONE+T)
  103 IF(NEL.LT.12) GO TO 200
      IF(IIX(11).EQ.0) GO TO 104
      SHP(2,11)=-S2**PFIVE*(ONE+R)*(ONE+T)
      SHP(4,11)=S2*(ONE+R)*(ONE+T)
  104 SHP(4,14)=R2*(ONE+S)*(ONE+T)
      IF(NEL.LT.13) GO TO 200
      IF(IIX(14).EQ.0) GO TO 105
      SHP(1,14)=-R2*PFIVE*(ONE+S)*(ONE-T)
      SHP(4,14)=R2*(ONE+S)*(ONE-T)
  105 SHP(4,15)=S2*(ONE-R)*(ONE-T)
      IF(NEL.LT.14) GO TO 200
      IF(IIX(15).EQ.0) GO TO 106
      SHP(2,15)=-S2**PFIVE*(ONE-R)*(ONE-T)
      SHP(4,15)=S2*(ONE-R)*(ONE-T)
  106 SHP(4,16)=R2*(ONE-S)*(ONE-T)
      IF(NEL.LT.15) GO TO 200
      IF(IIX(16).EQ.0) GO TO 107
      SHP(1,16)=-R2*PFIVE*(ONE-S)*(ONE-T)
      SHP(4,16)=R2*(ONE-S)*(ONE-T)
  107 SHP(4,17)=T2*(ONE+R)*(ONE+S)
      IF(NEL.LT.16) GO TO 200
      IF(IIX(17).EQ.0) GO TO 108
      SHP(3,17)=-T2*(ONE+R)*(ONE+S)
      SHP(4,17)=T2*(ONE+R)*(ONE+S)
  108 SHP(4,18)=T2*(ONE-R)*(ONE+S)
      IF(NEL.LT.17) GO TO 200
      IF(IIX(18).EQ.0) GO TO 109
      SHP(3,18)=-T2*(ONE-R)*(ONE+S)
      SHP(4,18)=T2*(ONE-R)*(ONE+S)
  109 SHP(4,19)=T2*(ONE-R)*(ONE-S)
      IF(NEL.LT.18) GO TO 200
      IF(IIX(19).EQ.0) GO TO 110
      SHP(3,19)=-T2*(ONE-R)*(ONE-S)
      SHP(4,19)=T2*(ONE-R)*(ONE-S)
  110 SHP(4,19)=-T*PFIVE*(ONE-R)*(ONE-S)
      IF(NEL.LT.19) GO TO 200
      IF(IIX(19).EQ.0) GO TO 111
      SHP(3,19)=-T*(ONE-R)*(ONE-S)
```

```
111   SHP(4,19) = T2*(ONE-R)*(ONE-S)
      IF(NEL.LT.20) GO TO 200
      IF(IX(1).EQ.0) GO TO 112
      SHP(1,20) = -T2*(ONE-S)
      SHP(2,20) = -T*PFIVE*(ONE+R)
      SHP(3,20) = -T*PFIVE*(ONE+R)*(ONE-S)
      SHP(4,20) = T2*(ONE+R)*(ONE-S)
      IF(NEL.LT.21) GO TO 200
112 C
C.... INTERIOR NODE (LAGRANGIAN)
      IF(IX(1).EQ.0) GO TO 200
      SHP(1,21) = -32.0D0*R*S2*T2
      SHP(2,21) = -32.0D0*T*R2*T2
      SHP(3,21) = -32.0D0*T*R2*S2
      SHP(4,21) = 64.0D0*R2*S2*T2
C
C.... CORRECT EDGE NODES FOR INTERIOR NODE (LAGRANGIAN)
      DO 114 J=1,4
      DO 113 I=1,8
113   SHP(J,I) = SHP(J,I) - 0.125D0*SHP(J,21)
      DO 114 I=9,20
114   IF(IX(I).NE.0) SHP(J,I) = SHP(J,I) - 0.25D0*SHP(J,21)
C
C.... CORRECT CORNER NODES FOR PRESENCE OF MIDSIDE NODES
200   K = 12
      DO 202 I = 1,4
      I4 = I + 4
      K4 = I + 8
      L  = I + 4
      L4 = L + 4
      DO 201 J = 1,4
      SHP(J,I ) = SHP(J,I ) - PFIVE*(SHP(J,K )+SHP(J,L )+SHP(J,M))
201   SHP(J,I4) = SHP(J,I4) - PFIVE*(SHP(J,K4)+SHP(J,L4)+SHP(J,M))
202   K=L
      RETURN
      END
C
      SUBROUTINE PGAUSS(GF,WF,NGP,L)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C.... GAUSS POINTS AND WEIGHTS
C
      DIMENSION GPWF(7,6)
      DATA GPWF/0.0D0,2.0D0,0.0D0,1.0D0,.55555555556D0,.34785484513745400,
     1.23692688505618900,1713244923791700D0,.5772501896260D0,.57773D0,
     2.26591926826D0,1.0D0,.88888888888889D0,.6521451548625460D0,.4786286700,
```

```
      349936CO ..360761573C481350C..7745966624148360,0.0D0,-.7745966624..
      414C83CO.,C:557269100..65214515482254460C..568888856D0,.4:..
      567513933C2:451C4531361C:1156405300..347854845C435848D0:-.3399936ECC
      6584846CO:.5861136310.-.8061178459384664DO.:53846881C:.470436ECC
      7524645:105C681798459304664DO..536426885026880+:36076157::-
      8538461350C-.1C45683CO...50617984513C435C4203152D0..2366245C0..3607617Ec
      930481313CC2:4691520.-.14203152DO6866619209386461203152C0..1608197Ec
      1122444:2375170CO/
      1224452=NGF+1
      NWF = NGF+1
      GP = GP#F(L,NGP)
      WF = GP#F(NWF,L)
      RETURN
      ENC

C....
C..1  SUBRCUTINE BCOND3 (UL,XL,IX,S,P,NDF,NDM,NST,NSBC,NGP,IFLAG)
      IMPLICIT REAL*8 (A-F,C-Z)
      COMMCN /CCATA/ O,HEAC,CM,ACT,IEL,NEL
      COMMCN /VELCATA/ LL,XL(4),IX(24),CT,IEL:NEL
      DIMENSICN KEL(1),XL(24),IX(1),S(NST,1),P(1),SHP(4,21)
      DIMENSICN KEL(50),KBCCND(50),KSURF(50),PROPE(50,2)
      GO TC (1,2),IFLAG

C.... INPUT SURFACE BOUNCARY CCNDITIONS
      WRITE(E,2CCO) O,HEAC
      KBMAX=0
      DO 10 I=1,NSBC
      REAC (E,1CCO) KEL(I),KBCCAD(I),KSURF(I),PRCFE(I,1),PROPB(I,2)
      KBMAX= MAXO(KBMAX,KBCCNC(I))
   10 WRITE(E,2001) KEL(I),KBCCNC(I),KSURF(I),PROPB(I,1),PROPB(I,2)
      IF(KBMAX.GE.4)CALL CCNV (T,COEFF,1)
      RETUFN

C.... ADD SURFACE BOUNDARY CONCITIONS CONTIBUITIONS
C.... TO CCNCUCTIVITY MATRIX ANC LOAD VECTOR
      DO 30 LL=1,NSBC
   2  IF(KEL(LL).NE.N) GO TO 3C
      KBT = KBCCNC(LL)
      CCNST =PROPE(LL,1)
      TEMP =PRCPB(LL,2)
      KBSAES = IABS(KBS)
      C3=3C CFLCAT(KES/KESAES)
      DO II=1,NGP
      CALL PGAL:S (C1,W1,NGP,II)
      DO 30 JJ=1,NGP
      CALL PGAL:S (C2,W2,NGP,JJ)
```

```
      HG = H1*H2
C.... SELECT CORRECT INTEGRATION POINTS
   21 GO TO (21,22,23),KBSABS
      RG = C1
      SG = C1
      TG = C1
      GO TO 24
   22 RG = C1
      SG = C1
      TG = C1
      GO TO 24
   23 RG = C1
      SG = C1
      TG = C1
   24 CALL SHAP3D (RG,SG,TG,XL,SHP,DUM,NDM,NEL,IX,.TRUE.)
      CALL JACEE3 (XSJ,KBSABS)
      DS = HG*XSJ
      COEFF = CCNST
      GO TO (25,26,27,28),KBT
C.... FLUX BCUNDARY CONDITION
   25 DO 251 L=1,NEL
  251 P(L) = F(L) - COEFF*SHP(4,L)*DS
      GO TO 30
C.... RADIATION BOUNDARY CONDITION
   27 T = C.CC
      DO 271 I=1,NEL
  271 T = T + SHP(4,I)*UL(I)
      COEFF = CCEFF*(T*T + TEMP*TEMP)*(T + TEMP)
      GO TO 26
C.... CONVECTION BOUNDARY CONDITION (TEMP. DEPENDENT)
   28 T = C.CC
      DO 281 I=1,NEL
  281 T = T + SHP(4,I)*UL(I)
      CALL CCNV(T,COEFF,2)
C.... CONVECTION BCUNDARY CONDITION
   26 DO 261 J=1,NEL
      A = SHP(4,J)*DS
      P(J) = F(J) + COEFF*TEMP*A
      DO 261 I = 1,NEL
  261 S(I,J) = S(I,J) + CCEFF*SHP(4,I)*A
   30 CONTINUE
      RETURN
 1000 FORMAT (3I5,2F10.0)
```

153

```
2000  FORMAT (A1,2CA4,//5X,12HSURFACE B.C./         TEMPERATURE)
     15X,52H   ELEM   B.C.,   SURF PROPERTY    VALUE
2001  FORMAT (5X,3(I5,2X),2(1PG15.7,2X))
      END

C
C.... CALCULATE CONDUCTIVITY (Z DIRECTION)
C
      SUBROUTINE PKZ (T,C,IFLAG)
      IMPLICIT REAL*8 (A-H,C-Z)

      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      DIMENSION TEMP(50),COEF(50)
      IF(IFLAG.EC.1) WRITE (6,2000) C,HEAD
      CALL TABLE (TEMP,COEF,T,C,NDATA,IFLAG)
      RETURN
2000  FORMAT(A1,2CA4,//5X,26HCONDUCTIVITY DATA (Z DIR.)//,
     1 5X,30H       TEMPERATURE          KY )
      END

      SUBROUTINE JACBB3 (XSJ,KS)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C.... COMPUTE SURFACE JACOBIAN DETERMINANT
C
      COMMON /JAC3S/ XS
      DIMENSION XS(3,3)
      GO TO (1,2,3),KS
1     C1=XS(2,3)*XS(3,3) - XS(2,3)*XS(3,2)
      C2=XS(2,1)*XS(3,1) - XS(2,2)*XS(3,3)
      C3=XS(2,2)*XS(3,2) - XS(2,1)*XS(3,1)
      GO TO 10
2     D1=XS(1,1)*XS(3,3) - XS(1,3)*XS(3,1)
      D2=XS(1,3)*XS(3,2) - XS(1,2)*XS(3,3)
      D3=XS(1,2)*XS(3,1) - XS(1,1)*XS(3,2)
      GO TO 10
3     D1=XS(1,1)*XS(2,2) - XS(1,2)*XS(2,1)
      D2=XS(1,3)*XS(2,3) - XS(1,1)*XS(2,2)
      D3=XS(1,2)*XS(2,1) - XS(1,3)*XS(2,3)
10    XSJ=CSQRT(D1*D1 + C2*C2 + D3*D3)
      RETURN
      END
```

LIST OF REFERENCES

1. Arpaci, V. S., <u>Conduction Heat Transfer</u>, Addison-Wesley Publishing Company, 1966.

2. Zienkiewicz, O. C. and Parekh, C. J., "Transient Field Problems: Two-Dimensional and Three-Dimensional Analysis by Isoparametric Finite Elements," <u>International Journal for Numerical Methods in Engineering</u>, v. 2, p. 61-71, January-March 1970.

3. Lew, G. T., <u>A Three Dimensional Solution of the Transient Field Problem Using Isoparametric Finite Elements</u>, M.S.M.E. Thesis, Naval Postgraduate School, Monterey, 1972.

4. Zienkiewicz, O. C., <u>The Finite Element Method</u>, 3rd ed., McGraw-Hill, 1977.

5. Bathe, K. J. and Wilson, E. L., <u>Numerical Methods in Finite Element Analysis</u>, Prentice-Hall, 1976.

6. Liniger, W., "Optimization of a numerical integration method for stiff systems of ordinary differential equations," <u>IBM Research Report RC2198</u>, 1968.

7. Wood, W. L., "On the Zienkiewicz Three- and Two-Time-Level Schemes Applied to the Integration of Parabolic Equations," <u>International Journal for Numerical Methods in Engineering</u>, v. 12, p. 1717-1726, 1978.

8. Lees, M., "A Linear Three-Level Difference Scheme for Quasilinear Parabolic Equations," <u>Mathematics of Computation</u>, v. 20, p. 516-622, 1966.

9. Hogge, M. A., <u>A Survey of Direct Integration Procedures for Nonlinear Transient Heat Transfer</u>, presented at the International Conference on Numerical Methods in Thermal Problems, University College, Swansea, U. K., 2-6 July, 1979.

10. Wood, W. L. and Lewis R. W., "A Comparision of Time Marching Schemes for the Transient Heat Conduction Equation," <u>International Journal for Numerical Methods in Engineering</u>, v. 9, p. 679-689, 1975.

11. Gresho, P. M. and Lee, R. L., <u>Don't Supress the Wiggles - They're Telling You Something!</u>, presented at the

Symposium on Finite Element Methods for Convection Dominated Flows, ASME Winter Annual Meeting, New York, 2-7 December, 1979.

12. Haji-Sheikh, A. and Sparrow, E. M., "The Solution of Heat Conduction Problems by Probability Methods," <u>Journal of Heat Transfer</u>, v. 89, p. 121-131, May 1967.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Documentation Center 2
   Cameron Station
   Alexandria, Virginia 22314

2. Library, Code 0142 2
   Naval Postgraduate School
   Monterey, California 93940

3. Department Chairman, Code 69 1
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, California 93940

4. Professor Gilles Cantin, Code 69Ci 10
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, California 93940

5. Professor O. C. Zienkiewicz, Code 69Zw 1
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, California 93940

6. 1° Ten. Jorge Bettencourt 3
   Direcção Do Serviço De Instrução
   Ministério Da Marinha
   Lisboa, PORTUGAL

7. Professor Paul F. Pucci, Code 69Pc 1
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, California 93940

8. Associate Professor Matthew D. Kelleher 1
   Code 69Kk
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, California 93940

9. Associate Professor David Salinas, Code 69Zc 1
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, California 93940

10. Associate Professor Richard H. Franke 1
    Code 53Fe
    Department of Mathematics
    Naval Postgraduate School
    Monterey, California 93940

11. Professor R. E. Newton, Code 69Ne   1
     Department of Mechanical Engineering
     Naval Postgraduate School
     Monterey, California 93940

12. Professor A. P. Boresi, Code 69Bh   1
     Department of Mechanical Engineering
     Naval Postgraduate School
     Monterey, California 93940

13. LCDR Lael R. Easterling, USN   1
     4243 N.W. 54th St.
     Oklahoma City, Oklahoma 73112

14. Georges Verchery   1
     Départment de Génie Mécanique
     Université de Technologie
     60200 Compiègne, FRANCE

15. Jean Louis Armand   1
     Institut de Recherche Pour la
     Construction Navale
     3 Avenue de Grand Champ
     78230 le Pecq, FRANCE

16. Prof. K. J. Bathe   1
     Mechanical Engineering Department
     M.I.T.
     77 Massachusetts Avenue
     Cambridge, Massachusetts   02139

17. William J. Dodge   1
     Oak Ridge National Laboratory
     Building 9204-1 Box Y
     Oak Ridge, Tennessee   36830

18. Jack Tree   1
     Air Research Manufacturing Co.
     402 South 36th Street
     P.O. Box 5217
     Phoenix, Arizona   85010

19. Prof. Edward L. Wilson   1
     Structural Engineering Division
     Civil Engineering Department
     University of California (Berkeley)
     Berkeley, California 94720

20. Dr. William J. Stronge (Code 603)   1
     Naval Weapons Center
     China Lake, California   93955

21. J. E. Serpanos   1
     Code 3162
     Naval Weapons Center
     China Lake, California 93955

22. Dr. Jean Louis Batoz   1
     Département de Génie Mécanique,
     Université de Technologie,
     60200 Compiègne, FRANCE

23. Professor Guri Dhatt   1
     Centre Technique de l'Informatique
     Université Laval
     Québec, Prov. de Québec
     CANADA   GIK 7P4

24. John Fairbanks   1
     Department of Energy
     Division of Power Systems
     20 Massachusetts Avenue, NW
     Washington, DC   20545

25. Dr. Gilbert Touzot   1
     Centre d'Informatique
     Université de Technologie
     60206 Compiègne, FRANCE

26. R. A. Langworthy   1
     Applied Technology Laboratories
     U.S. Army Research and Technology
      Laboratory
     Fort Eustis, Virginia   23604

27. E. M. Lenoe   1
     Army Materials & Mechanic Research
      Center
     Arsenal Street
     Watertown, Massachusetts   02172

28. Dr. Paris Genalis   1
     Naval Ship Research and Development Center
     Bethesda, Maryland   20084

29. C. Miller   1
     Naval Sea Systems Command
     Department of the Navy
     Washington, DC   20362

30. Code SEC 6734   1
     Naval Ship Engineering Center
     Philadelphia Division
     Philadelphia, Pennsylvania   19112

31. A. M. Diness (Code 471)   1
     Department of the Navy
     Office of Naval Research
     Arlington, Virginia   22217

|		
32.	Ray M. Standahar Office of Secretary of Defense DDR&E 3D1089 Pentagon Washington, DC 20301	1
33.	R. Rice (Code 6360) Naval Research Laboratory Washington, DC 20375	1
34.	E. Van Reuth Defense Advanced Research Projects Agency 1440 Wilson Boulevard Arlington, Virginia 22209	1
35.	I. Machlin Naval Air System Command Department of the Navy Washington, DC 20361	1
36.	B. Probst, MS 49-3 NASA-Lewis Research Center 21000 Brookpark Road Cleveland, Ohio 44135	1
37.	Mr. C. P. Blankenship, MS 105-1 NASA-Lewis Research Center 21000 Brookpark Road Cleveland, Ohio 44135	1
38.	Dr. H. Graham/AFML/LLM Department of the Air Force Air Force Materials Laboratory Wright-Patterson Air Force Base, Ohio 45433	1
39.	Mr. George Strong DCASMA, Phoenix 3800 North Central Avenue Phoenix, Arizona 85012	1
40.	S. Freiman (Code 6363) Naval Research Laboratory Washington, DC 20375	1
41.	AFML/LLM/N. M. Geyer Air Force Materials Laboratory Wright-Patterson Air Force Base, Ohio 45433	1
42.	S. Wiederhorn National Bureau of Standards Washington, DC 20234	1

43. LT J. H. Preisel, Jr., USN   1
     922 Bernard Road
     Peekskill, New York   10566

44. Allan F. Greiner   1
     United Technologies Research Center
     East Hartford, Connecticut   06108

45. LCDR L. B. Elliott   1
     Code 338.14
     Long Beach Naval Shipyard
     Long Beach, California   90822

46. Martin A. Krenzke   1
     David W. Taylor Naval Ship
      Research and Development Center
     A236, Bldg. 19
     Carderock Laboratory
     Bethesda, Maryland   20084

47. Professor Robert L. Taylor   1
     Department of Civil Engineering
     University of California
      at Berkeley
     Berkeley, California   94720